Army Research Laboratory

Multiplicative Quaternion Extended Kalman Filtering for Nonspinning Guided Projectiles

by James M. Maley

ARL-TR-6503 **July 2013**

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer’s or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory
Aberdeen Proving Ground, MD 21005-5066

ARL-TR-6503 **July 2013**

Multiplicative Quaternion Extended Kalman Filtering for Nonspinning Guided Projectiles

James M. Maley
Weapons and Materials Research Directorate, ARL

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)* July 2013	**2. REPORT TYPE** Final	**3. DATES COVERED (From - To)** October 2012–December 2012

4. TITLE AND SUBTITLE

Multiplicative Quaternion Extended Kalman Filtering for Nonspinning Guided Projectiles

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

James M. Maley

5d. PROJECT NUMBER

AH80

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

U.S. Army Research Laboratory
ATTN: RDRL-WML-F
Aberdeen Proving Ground, MD 21005-5066

8. PERFORMING ORGANIZATION REPORT NUMBER

ARL-TR-6503

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The problem of estimating the attitude, velocity, and position states of a slow to nonspinning, gun-launched projectile is addressed through the use of an extended Kalman filter (EKF). The EKF is constructed using an error-state mechanization similar to those used in traditional inertial navigation applications, although the kinematics are simplified by assuming the Earth-fixed reference frames used for tactical applications are inertial. The advantages of using quaternions rather than Euler angles to represent projectile attitude are discussed, and the use of multiplicative quaternion error states is given detailed attention as this leads to fundamental differences from most other extended Kalman filter implementations that tend to assume additive error states. The measurements and heuristic information available for most projectile applications are incorporated into the EKF. Two sets of simulation results involving a direct-fire system and an indirect-fire system are presented.

15. SUBJECT TERMS

Kalman filter, EKF, projectile, navigation, state estimation, inertial, magnetometers

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** James M. Maley
a. REPORT Unclassified	**b. ABSTRACT** Unclassified	**c. THIS PAGE** Unclassified	UU	48	**19b. TELEPHONE NUMBER** (*Include area code*) 410-306-0814

Contents

List of Figures

List of Tables

Acknowledgments

The simulation results leverage the aerodynamic modeling work done by Dr. Frank Fresconi, and the simulink implementation of the projectile flight dynamics, flight control, and guidance laws by Dr. Fresconi, Dr. Mark Ilg, and me, for other research and development projects. I would also like to thank them for their helpful advice and comments.

1. Introduction

The primary motivation for this work is to create a general framework for estimating the attitude, velocity, and position states for roll-controlled, guided projectiles for which assumptions about the projectile dynamics may not be valid. Guided projectiles could potentially offer several strategic advantages to the Warfighter by reducing the miss distance, shaping the terminal trajectory, and/or homing to moving targets. While guidance solutions exist for both fin-stabilized and spin-stabilized spinning projectiles, nonspinning projectiles are able to achieve more lateral acceleration for conducting more effective maneuvers than their spinning counterparts (*1*). The challenge with nonspinning projectiles is that they don't benefit from the added observability provided from a predictable spin rate. The "slowly varying spin rate" assumption is often used to correct for sensor biases, either directly (*2*, *3*) or indirectly (*4*). As a simple example, consider the output of a radial magnetometer as depicted in figure 1. The spinning of the round makes it fairly easy to separate a magnetometer sensor bias from the roll angle with respect to the magnetic field vector because the output signal is periodic. This can either be accomplished with empirical routines such as peak finding, or with a linear state estimator by augmenting the projectile state vector with magnetometer bias states that are modeled as constants. If the projectile stops rolling, this continuous observability goes away, and the estimator must rely much more heavily on the process models to account for the uncertainty in attitude and sensor errors.

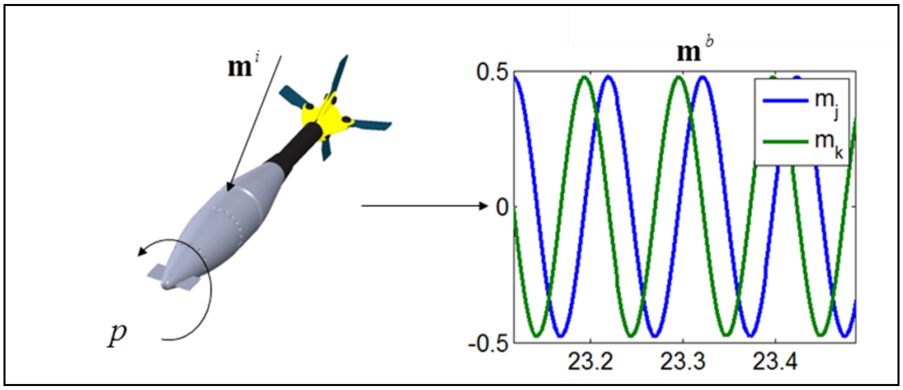

Figure 1. Projection of the magnetic field vector in inertial (*i*) coordinates into the body-fixed (*b*) frame as measured by magnetometers for a rolling projectile

To increase the fidelity of the attitude estimator process model, the propagation equations used in the prediction step of the Kalman filtering process are based on the output of an inertial measurement unit (IMU) rather than a projectile-specific dynamic model. While the dynamic model predictions may still be useful as a heuristic measurement, a dynamic model is usually an approximation of the true dynamics. Errors in the prediction quality are usually approximated by

Gaussian white noise acting as a random input to the dynamic system. Since this noise is only an artifact of mathematical convenience rather than a true physical phenomenon, it cannot be measured and is usually used as a tuning parameter. When using kinematics driven by IMU output as the process model, the statistics of the gyroscope and accelerometer noise are measureable, and can be easily incorporated into an extended Kalman filtering framework. A realistic IMU typically has its own sources of modeling error due to sensor calibration errors, non-linearity, cross-axis sensitivity, g-sensitivity, misalignment, etc., which can be either modeled or approximated by increasing the covariance of the process noise.

In the past, both dynamic models (*5*) and kinematic models (*6*) have been used as process models for projectile Kalman filters, as well as combinations of the two (*7*). There have been practical concerns with using an IMU on a projectile, the first of which is the availability of sensors that are capable of measuring the spin rate of a projectile. On a spin-stabilized projectile spinning in the neighborhood of 300 Hz, this is still the case. However, in recent years, micro-electromechanical system (MEMS) gyroscopes have been able to measure the spin-rates of fin-stabilized projectiles such as mortars, which typically spin at rates in the neighborhood of 20 Hz. It is assumed that any nonspinning projectile will not be de-spun from any rate higher than that of typical fin-stabilized projectiles, if indeed it needs to be de-spun at all. The other concern with using IMUs on gun-launched projectiles is the tradeoff between affordability, durability, and performance. Automotive-grade MEMS components have been used in the harsh gun-launch environment for years, and MEMS technology has generally demonstrated its survivability and affordability.

There are still some challenges with MEMS components, however. In the past, sensor bias changes have been observed following the launch event, and most inertial sensors either saturate or temporarily malfunction during the actual launch event although they recover milliseconds after leaving the gun barrel. This leads to an inertial navigation challenge that is particular to gun-launched munitions: the full or partial loss of initial conditions immediately following gun launch. The final concern is sensor quality. The environmental and cost restrictions prohibit the use of most navigation grade or even tactical-grade IMUs used in other applications. However, the flight times of guided munitions are very short, so random walk due to relatively poor noise performance is not as much of a concern as determining calibration and initial condition errors imposed by gun launch. In addition, the MEMS industry is still making progress in improving the noise performance, linearity, temperature stability, cross-axis sensitivity, and vibration rejection of their products, and will likely continue to do so for some time. Therefore, at the very minimum a framework should be in place for evaluating these components for projectile state estimation as they improve.

The choice of attitude representation is an important one. Both quaternions and direction cosine matrices (DCMs) are common in aerospace applications. Quaternions seem to be used in the majority of attitude estimators, because they have fewer elements to store in memory than a DCM, and they are easier to normalize. Euler angles typically don’t see much use outside of projectile applications, because Euler angles can become discontinuous when representing certain orientations and their propagation equations can become indeterminate (*8*). Since gun-launched projectiles generally don't experience a wide range of heading angles, this has not a primary concern. The other major advantage of quaternions (or DCMs) over Euler angles is their propagation equations are linear with respect to the quaternion and only depend on the IMU's angular velocity. This enables much more effective use of the angular rate sensors, as described in section 3 of this report.

A secondary motivation for writing this report is to provide a detailed explanation of multiplicative quaternion error-state Kalman filter mechanizations. Although this form of the Kalman filter has been used in space applications since the 1960s (*9*) and many other applications (*10*), clear documentation, especially in terms of guided projectiles, has been difficult to find in the literature. Section 2 discusses notation, coordinate systems, and some preliminary quaternion information that should make this report more readable. Section 4 presents a derivation of the error state kinematic equations that parallels those presented in Crassidis and Junkins (*11*) and Roumeliotis et al. (*12*), although a different convention for the quaternion definition is used to be consistent with the conventions used in MATLAB* software and Titterton and Weston (*8*) and Stevens and Kerce (*13*).

The position and velocity error state kinematics are included for completeness, although in a very simplified form. In the scope of this work, the local Earth-fixed navigation system is treated as inertial, so many aspects of inertial navigation such as Coriolis acceleration, Schuler tuning, and plumb-bob gravity are not covered here.

Section 5 discusses the measurement of the process noise. The measurements commonly available to guided projectiles and the mapping of the error states to the residuals, the differences between the measured sensor outputs and predicted ones, are discussed in section 6. Section 7 discusses how all of the error state modeling can be implemented in a Kalman filter running on a microcontroller with processing delays. Two different guided projectile missions are simulated to examine the performance of a Kalman filter that estimates the attitude, velocity, position, gyro bias, accelerometer bias, and magnetometer bias errors. The first mission is a short range, direct fire mission where a nonspinning round is controlled and guided to a moving target. The second mission is a mortar that is de-spun near apogee and maneuvers to a ground-based target. The simulations and filter performance are discussed in section 8. Finally, section 9 discusses conclusions and the direction of future research.

* MATLAB is a registered trademark of MathWorks, Inc., Natick, MA.

2. Conventions

Vectors are defined by lowercase bold script. The terms reference frame and coordinate system are used interchangeably, and all coordinate systems consist of three orthonormal basis vectors; for example, coordinate system a consists of the vectors $\mathbf{x}_a, \mathbf{y}_a, \mathbf{z}_a$. A superscript denotes the coordinate system in which the coordinates of the vector are viewed; for example, $\mathbf{v}^a$ is the coordinate of the vector $\mathbf{v}$ in the coordinate system a. There are two main reference frames used in this work. The first is the inertial reference (i-frame). A major simplification used throughout this work is that any reference frame rigidly fixed to the surface of the Earth can be treated as an inertial reference frame for the application to gun-launched projectiles. The inertial frame used here has the z-axis pointing toward the center of the Earth, the x-axis pointed along a level line from the gun toward the initial target location, and the y-axis pointed to the right of the line of fire. The second coordinate system used is the body-fixed reference (b-frame), which is rigidly attached to the projectile. For the purposes of inertial navigation, it is assumed that the origin of the body frame is located at the location of the accelerometers in the IMU. This prevents the need to account for centripetal and tangential acceleration terms in the integration process. However, the velocity and position states obtained will be those of the IMU and not those of the center of gravity (CG) of the projectile. The two coordinate systems are displayed in figure 2.

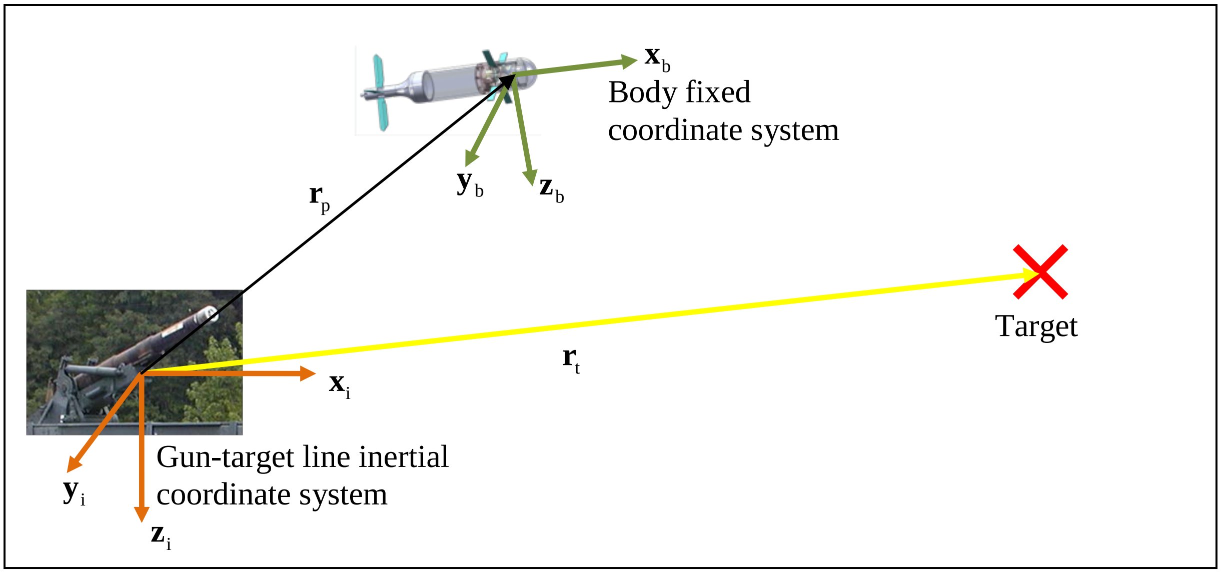

Figure 2. Coordinate system definitions.

To relate the (b-frame) to the (i-frame), we will use quaternion arithmetic. The definition of a quaternion used here has the scalar component as the first element, and the vector component as the second through fourth element, i.e.,

$$\mathbf{q} = \begin{Bmatrix} q_1 \\ \mathbf{q}_{2:4} \end{Bmatrix} . \tag{1}$$

The rotation vector $\boldsymbol{\mu}$ can also be used to parameterize the quaternion as follows:

$$\begin{Bmatrix} q_1 \\ \mathbf{q}_{2:4} \end{Bmatrix} = \begin{Bmatrix} \cos\left(\dfrac{\|\boldsymbol{\mu}\|}{2}\right) \\ \dfrac{\boldsymbol{\mu}}{\|\boldsymbol{\mu}\|}\sin\left(\dfrac{\|\boldsymbol{\mu}\|}{2}\right) \end{Bmatrix} . \tag{2}$$

The quaternion product, denoted $\otimes$, of any two quaternions $\mathbf{p}$ and $\mathbf{q}$ is given by (*14*):

$$\mathbf{p} \otimes \mathbf{q} = \begin{Bmatrix} p_1 q_1 - \mathbf{p}_{2:4} \cdot \mathbf{q}_{2:4} \\ p_1 \mathbf{q}_{2:4} + q_1 \mathbf{p}_{2:4} + \mathbf{p}_{2:4} \times \mathbf{q}_{2:4} \end{Bmatrix} . \tag{3}$$

The quaternion inverse is defined as the scalar component with the negative of the vector component:

$$\mathbf{q}^{-1} = \begin{bmatrix} q_1 \\ -\mathbf{q}_{2:4} \end{bmatrix} . \tag{4}$$

This quaternion inverse multiplied by the quaternion equals the identity quaternion:

$$\mathbf{q}^{-1} \otimes \mathbf{q} = \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix} . \tag{5}$$

Converting a vector from the inertial frame to the body frame can be accomplished with quaternion rotation:

$$\begin{bmatrix} 0 \\ \mathbf{v}^b \end{bmatrix} = \mathbf{q}^{-1} \otimes \begin{bmatrix} 0 \\ \mathbf{v}^i \end{bmatrix} \otimes \mathbf{q} . \tag{6}$$

The same rotation can also be accomplished by converting the quaternion into the direction cosine matrix:

$$\mathbf{v}^b = \mathbf{C}_i^b(\mathbf{q})\mathbf{v}^i , \tag{7}$$

where:

$$\mathbf{C}_i^b\left(\mathbf{q}\right)=2\mathbf{q}_{2:4}\mathbf{q}_{2:4}^T+\mathbf{I}_{3x3}\left(q_1^2-\mathbf{q}_{2:4}^T\mathbf{q}_{2:4}\right)-2q_1\left[\mathbf{q}_{2:4}\times\right]. \tag{8}$$

The general rotation matrix operator $\mathbf{C}_i^b\left(\mathbf{q}\right)$ reads “transformation from i to b frame coordinates parameterized by the quaternion $\mathbf{q}$.” The $\left[\bullet\times\right]$ operator denotes the skew-symmetric matrix of a vector $\mathbf{v}\in\square^3$, so that multiplying a vector times the matrix is equivalent to taking the cross product of the two vectors, i.e.,

$$\left[\mathbf{v}\times\right]=\begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix}, \tag{9}$$

so that

$$\left[\mathbf{v}\times\right]\mathbf{w}=\mathbf{v}\times\mathbf{w}\,. \tag{10}$$

3. IMU Integration With Quaternions

Integrating the gyro outputs to obtain the projectile attitude is a prerequisite for integrating acceleration to obtain the velocity and position states. The kinematic equations for a quaternion are given by:

$$\dot{\mathbf{q}}=\frac{1}{2}\mathbf{q}\otimes\begin{Bmatrix} 0 \\ \boldsymbol{\omega}_{ib}^b \end{Bmatrix}. \tag{11}$$

The vector $\boldsymbol{\omega}_{ib}^b$ consists of the body-fixed coordinates of the angular velocity of the body frame with respect to the inertial reference frame. Equation 3 can be used to reformulate equation 11 as a matrix multiplication:

$$\begin{aligned}\dot{\mathbf{q}}&=\frac{1}{2}\begin{bmatrix} 0 & -\boldsymbol{\omega}_{ib}^{b\,T} \\ \boldsymbol{\omega}_{ib}^b & -\left[\boldsymbol{\omega}_{ib}^b\times\right] \end{bmatrix}\mathbf{q}\\ &=\frac{1}{2}\boldsymbol{\Omega}\left(\boldsymbol{\omega}_{ib}^b\right)\mathbf{q}\end{aligned}. \tag{12}$$

It is clear that the quaternion propagation equations are linear with respect to the quaternion. The algorithm presented here is identical to that presented in reference 8 for computing the solution to equation 12 between time steps in a sampled system, although the explanation differs slightly. For the remainder of this section, the reference frame notation $\boldsymbol{\omega}_{ib}^b$ will be replaced by iteration notation $\boldsymbol{\omega}_k$, meaning the value of $\boldsymbol{\omega}_{ib}^{b}$ sampled at time k. In this case the solution to equation 12 is given by:

$$\mathbf{q}_{k+1} = \exp\left(\frac{1}{2}\boldsymbol{\Omega}\left(\bar{\boldsymbol{\omega}}\right)\Delta t\right)\mathbf{q}_k . \tag{13}$$

The value of $\bar{\boldsymbol{\omega}}$ is usually taken to be:

$$\bar{\boldsymbol{\omega}} = \frac{\boldsymbol{\omega}_k + \boldsymbol{\omega}_{k-1}}{2} . \tag{14}$$

The incremental rotation vector can be approximated as[*] $\boldsymbol{\sigma} = \bar{\boldsymbol{\omega}}\Delta t$. For a given rotation vector, the exact solution for the matrix exponential in equation 13 is given by (*8*):

$$\exp\left(\frac{1}{2}\boldsymbol{\Omega}\left(\boldsymbol{\sigma}\right)\right) = \left[\begin{array}{c:c} \cos\left(\left\|\boldsymbol{\sigma}\right\|/2\right) & \dfrac{\sin\left(\left\|\boldsymbol{\sigma}\right\|/2\right)}{2}\boldsymbol{\sigma}^{T} \\ \hdashline \dfrac{\sin\left(\left\|\boldsymbol{\sigma}\right\|/2\right)}{2}\boldsymbol{\sigma} & \cos\left(\left\|\boldsymbol{\sigma}\right\|/2\right)\mathbf{I}_{3x3} - \dfrac{\sin\left(\left\|\boldsymbol{\sigma}\right\|/2\right)}{2}\left[\boldsymbol{\sigma}\times\right] \end{array}\right]. \tag{15}$$

The velocity dynamics are a function of the specific force measured by the accelerometers $\mathbf{f}^{b}$, the gravity vector $\mathbf{g}^{i}$, and the quaternion:

$$\dot{\mathbf{v}}^{i} = \mathbf{C}_{b}^{i}\left(\mathbf{q}\right)\mathbf{f}^{b} + \mathbf{g}^{i} . \tag{16}$$

Using equations 13–15, it is possible to predict what the quaternion will be at the current time step based on angular rate sensor data. Velocity and position are estimated via simple trapezoidal integration, which requires the quaternion at time k to be estimated first. The velocity update is given by:

$$\mathbf{v}_k = \left(\frac{\mathbf{C}_{b}^{i}\left(\mathbf{q}_k\right)\mathbf{f}_{k}^{b} + \mathbf{C}_{b}^{i}\left(\mathbf{q}_{k-1}\right)\mathbf{f}_{k-1}^{b}}{2} + \mathbf{g}^{i}\right)\Delta t + \mathbf{v}_{k-1} . \tag{17}$$

The position update is then given by:

$$\mathbf{r}_{k}^{i} = \frac{\mathbf{v}_{k}^{i} + \mathbf{v}_{k-1}^{i}}{2}\Delta t + \mathbf{r}_{k-1}^{i} . \tag{18}$$

[*] There exist higher-fidelity representations of the rotation vector. Bortz (*15*) derives the rotation vector, its dynamics, and its relation to the DCM from geometric arguments. It was shown that the rotation vector can be integrated at a high integration rate and then used to update the DCM only when needed, significantly increasing algorithm efficiency. Since then, particularly in Savage (*16*), several algorithms have been created for computing the rotation vector on digital processors, and how the rotation vector relates to the DCM and quaternion attitude representation. The advantage of using the trapezoidal approximation used here is that calibration corrections can easily be applied after the inertial sensors have been sampled as discussed in more detail in section 7, without needing to retain several samples in memory. The disadvantage is that the approximation will necessitate a faster KF update rate.

This algorithm assumes that the specific force in the inertial frame is fairly constant over the update interval. The inertial specific force $\mathbf{f}^{i} = \mathbf{C}_b^i(\mathbf{q})\mathbf{f}^b$, can be computed in terms of the projectile's acceleration $\dot{\mathbf{v}}_{cg}^{i}$ and the location of the CG relative to the body frame $\mathbf{r}_{cg}^{b}$:

$$\mathbf{C}_b^i(\mathbf{q})\mathbf{f}^b = \dot{\mathbf{v}}_{cg}^i - \mathbf{C}_b^i[\boldsymbol{\omega}\times][\boldsymbol{\omega}\times]\mathbf{r}_{cg}^b - \mathbf{C}_b^i[\dot{\boldsymbol{\omega}}\times]\mathbf{r}_{cg}^b - \mathbf{g}^i . \tag{19}$$

Full derivation is given in appendix A.

On a guided projectile, the acceleration of the CG does not usually change dramatically since most projectiles have limited control authority. Mounting the accelerometer triad on the spin axis is strongly advisable, since centrifugal accelerations can be quite large. Algorithms in literature typically don't use a trapezoidal integration scheme at all but use a combination of incremental velocity, rotation correction, and dynamic integral terms, and potentially more advanced sculling and scrolling compensation algorithms (*8*, *16*). However, for our purposes the algorithm provides a method of post applying calibration corrections and is computationally tractable.

4. Error State Formulation

The integration of the quaternion, velocity, and position states is in reality driven by non-perfect gyroscopes and accelerometers. Therefore, the integrated states are only estimations of the true states. Terms that are estimates are denoted by the ^ over script. The estimated state dynamics are given by:

$$\dot{\hat{\mathbf{q}}} = \frac{1}{2}\hat{\mathbf{q}} \otimes \begin{Bmatrix} 0 \\ \hat{\boldsymbol{\omega}}_{ib}^{b} \end{Bmatrix}, \tag{20}$$

$$\dot{\hat{\mathbf{v}}}^i = \mathbf{C}_b^i(\hat{\mathbf{q}})\hat{\mathbf{f}}^b + \hat{\mathbf{g}}^i , \tag{21}$$

and

$$\dot{\hat{\mathbf{r}}}^i = \hat{\mathbf{v}}^i . \tag{22}$$

From this point forward, we will drop the super and subscripts for simplicity. The IMU outputs estimated angular velocity and specific force vectors, which are defined by:

$$\boldsymbol{\omega} = \hat{\boldsymbol{\omega}} + \delta\boldsymbol{\omega} . \tag{23}$$

$$\mathbf{f} = \hat{\mathbf{f}} + \delta\mathbf{f} . \tag{24}$$

The perturbation terms $\delta\boldsymbol{\omega}$ and $\delta\mathbf{f}$ can be parameterized by any noise and sensor calibration errors substantial enough to affect the navigation solution.

The true quaternion $\mathbf{q}$ can be described as the quaternion multiplication of the estimated quaternion $\hat{\mathbf{q}}$ and a small error quaternion $\delta\mathbf{q}$:

$$\mathbf{q} = \hat{\mathbf{q}} \otimes \delta\mathbf{q} . \tag{25}$$

$$\delta\mathbf{q} = \hat{\mathbf{q}}^{-1} \otimes \mathbf{q} . \tag{26}$$

This description is not common in other extended Kalman filtering problems, which typically treat the error terms as additive, i.e., $\mathbf{q} = \hat{\mathbf{q}} + \delta\mathbf{q}$. The additive error assumption is used in most Kalman filtering texts such as Simon (*17*) and Gelb (*18*). Indeed, there are even several quaternion estimators that use this assumption (*10*). Adding two unit quaternions together does not produce another unit quaternion, which creates a problem that is normally dealt with by gratuitous renormalization. It is possible, but quite complicated, to correctly account for these computational aspects in the error covariance matrix (*19*). The multiplicative error formulation also has the physical interpretation of applying a small rotation correction. This can easily be shown in the formulation of the composite DCM. Using the conventions in this work, the total DCM is given by:

$$\mathbf{C}_i^b(\mathbf{q}) = \mathbf{C}(\hat{\mathbf{q}} \otimes \delta\mathbf{q}) = \mathbf{C}(\delta\mathbf{q})\mathbf{C}(\hat{\mathbf{q}}) . \tag{27}$$

Another benefit of the multiplicative error quaternion is that only three terms are needed to parameterize it instead of four. Since the error quaternion represents a small rotation, the rotation vector $\boldsymbol{\mu}$ from equation 2 has small magnitude which means that the scalar component of $\delta\mathbf{q}$ is approximately 1 and need not be estimated. The only downside is that the linearization of the processes and measurement equations cannot be accomplished simply by computing Jacobians.

The quaternion error state dynamics are produced by differentiating equation 26 with respect to time:

$$\delta\dot{\mathbf{q}} = \hat{\mathbf{q}}^{-1} \otimes \dot{\mathbf{q}} + \dot{\hat{\mathbf{q}}}^{-1} \otimes \mathbf{q} . \tag{28}$$

A substitution for $\dot{\hat{\mathbf{q}}}^{-1}$ can be generated from the fact that $\hat{\mathbf{q}}^{-1} \otimes \hat{\mathbf{q}}$ always produces the identity quaternion, which is constant.

$$\frac{d\left(\hat{\mathbf{q}}^{-1} \otimes \hat{\mathbf{q}}\right)}{dt} = \dot{\hat{\mathbf{q}}}^{-1} \otimes \hat{\mathbf{q}} + \hat{\mathbf{q}}^{-1} \otimes \dot{\hat{\mathbf{q}}} = \mathbf{0} . \tag{29}$$

$$\begin{aligned} \dot{\hat{\mathbf{q}}}^{-1} &= -\hat{\mathbf{q}}^{-1} \otimes \dot{\hat{\mathbf{q}}} \otimes \hat{\mathbf{q}}^{-1} \\ &= -\hat{\mathbf{q}}^{-1} \otimes \frac{1}{2}\hat{\mathbf{q}} \otimes \begin{Bmatrix} 0 \\ \hat{\boldsymbol{\omega}}_{ib}^{b} \end{Bmatrix} \otimes \hat{\mathbf{q}}^{-1} . \\ &= -\frac{1}{2}\begin{Bmatrix} 0 \\ \hat{\boldsymbol{\omega}}_{ib}^{b} \end{Bmatrix} \otimes \hat{\mathbf{q}}^{-1} \end{aligned} \tag{30}$$

Substituting equations 30 and 11 into equation 28 leads to:

$$\begin{aligned} \delta\dot{\mathbf{q}} &= \hat{\mathbf{q}}^{-1} \otimes \frac{1}{2}\mathbf{q} \otimes \left\{ \begin{matrix} 0 \\ \boldsymbol{\omega}_{ib}^{b} \end{matrix} \right\} - \frac{1}{2} \left\{ \begin{matrix} 0 \\ \hat{\boldsymbol{\omega}} \end{matrix} \right\} \otimes \hat{\mathbf{q}}^{-1} \otimes \mathbf{q} \\ \delta\dot{\mathbf{q}} &= \frac{1}{2}\delta\mathbf{q} \otimes \left\{ \begin{matrix} 0 \\ \boldsymbol{\omega} \end{matrix} \right\} - \frac{1}{2} \left\{ \begin{matrix} 0 \\ \hat{\boldsymbol{\omega}} \end{matrix} \right\} \otimes \delta\mathbf{q} \end{aligned} . \tag{31}$$

Using the definition of the estimated angular velocity (equation 20), this becomes:

$$\delta\dot{\mathbf{q}} = \frac{1}{2}\delta\mathbf{q} \otimes \left\{ \begin{matrix} 0 \\ \hat{\boldsymbol{\omega}} \end{matrix} \right\} - \frac{1}{2} \left\{ \begin{matrix} 0 \\ \hat{\boldsymbol{\omega}} \end{matrix} \right\} \otimes \delta\mathbf{q} + \frac{1}{2}\delta\mathbf{q} \otimes \left\{ \begin{matrix} 0 \\ \delta\boldsymbol{\omega} \end{matrix} \right\} . \tag{32}$$

The definition of quaternion multiplication can now be applied along with the fact that $\delta q_1 = 1$ to achieve further simplification:

$$\begin{aligned} \delta\dot{\mathbf{q}} &= \frac{1}{2} \left\{ \begin{matrix} -\delta\mathbf{q}_{2:4} \cdot \hat{\boldsymbol{\omega}} \\ \hat{\boldsymbol{\omega}} + \delta\mathbf{q}_{2:4} \times \hat{\boldsymbol{\omega}} \end{matrix} \right\} + \frac{1}{2} \left\{ \begin{matrix} \hat{\boldsymbol{\omega}} \cdot \delta\mathbf{q}_{2:4} \\ -\hat{\boldsymbol{\omega}} - \hat{\boldsymbol{\omega}} \times \delta\mathbf{q}_{2:4} \end{matrix} \right\} + \frac{1}{2} \left\{ \begin{matrix} -\delta\mathbf{q}_{2:4} \cdot \delta\hat{\boldsymbol{\omega}} \\ \delta\hat{\boldsymbol{\omega}} + \delta\mathbf{q}_{2:4} \times \delta\hat{\boldsymbol{\omega}} \end{matrix} \right\} \\ \left\{ \begin{matrix} 0 \\ \delta\dot{\mathbf{q}}_{2:4} \end{matrix} \right\} &= \left\{ \begin{matrix} 0 \\ -\hat{\boldsymbol{\omega}} \times \delta\mathbf{q}_{2:4} \end{matrix} \right\} + \frac{1}{2} \left\{ \begin{matrix} -\delta\mathbf{q}_{2:4} \cdot \delta\hat{\boldsymbol{\omega}} \\ \delta\hat{\boldsymbol{\omega}} + \delta\mathbf{q}_{2:4} \times \delta\hat{\boldsymbol{\omega}} \end{matrix} \right\} \end{aligned} . \tag{33}$$

Since the quaternion error states are assumed to be small in magnitude, the error state dynamics can be approximated by:

$$\delta\dot{\mathbf{q}}_{2:4} \cong -\hat{\boldsymbol{\omega}} \times \delta\mathbf{q}_{2:4} + \frac{1}{2}\delta\boldsymbol{\omega} . \tag{34}$$

At this point, it is convenient to replace the quaternion error states $\delta\mathbf{q}_{2:4}$ with a vector of small angles:

$$\boldsymbol{\alpha} = 2\delta\mathbf{q}_{2:4} . \tag{35}$$

The first convenience comes from removing the factor of ½ from equation 34:

$$\dot{\boldsymbol{\alpha}} \cong -\hat{\boldsymbol{\omega}} \times \boldsymbol{\alpha} + \delta\boldsymbol{\omega} . \tag{36}$$

The second convenience comes from the simplification of equation 27. Evaluating equation 8 with $\delta\mathbf{q}$ and assuming products of the terms in $\delta\mathbf{q}_{2:4}$ are approximately zero results in $\mathbf{C}_i^b\left(\delta\mathbf{q}\right) \cong \left[\mathbf{I} - 2\left[\delta\mathbf{q}_{2:4} \times\right]\right]$. Therefore, the DCM and its inverse can be expressed as:

$$\mathbf{C}_i^b\left(\mathbf{q}\right) \cong \left[\mathbf{I} - \left[\boldsymbol{\alpha} \times\right]\right] \mathbf{C}_i^b\left(\hat{\mathbf{q}}\right) . \tag{37}$$

$$\mathbf{C}_b^i\left(\mathbf{q}\right) \cong \mathbf{C}_b^i\left(\hat{\mathbf{q}}\right) \left[\mathbf{I} + \left[\boldsymbol{\alpha} \times\right]\right] . \tag{38}$$

The velocity and position error state dynamics are simpler to derive, since their error states are additive:

$$\delta\mathbf{v} = \mathbf{v} - \hat{\mathbf{v}}\,. \tag{39}$$

$$\delta\mathbf{r} = \mathbf{r} - \hat{\mathbf{r}}\,. \tag{40}$$

The velocity error states propagate according to:

$$\delta\dot{\mathbf{v}} = \dot{\mathbf{v}} - \dot{\hat{\mathbf{v}}}\,. \tag{41}$$

By utilizing equations 21, 24, 38, and 41, an expression for the velocity error state dynamics can be derived:

$$\delta\dot{\mathbf{v}} = \mathbf{C}_b^i\left(\mathbf{q}\right)\mathbf{f}^b + \mathbf{g} - \mathbf{C}_b^i\left(\hat{\mathbf{q}}\right)\hat{\mathbf{f}}^b + \hat{\mathbf{g}}\ \delta\dot{\mathbf{v}} = \mathbf{C}_b^i\left(\hat{\mathbf{q}}\right)\left[\mathbf{I} + \left[\boldsymbol{\alpha}\times\right]\right]\left(\hat{\mathbf{f}}^b + \delta\mathbf{f}\right) + \mathbf{g} - \mathbf{C}_b^i\left(\hat{\mathbf{q}}\right)\hat{\mathbf{f}}^b - \hat{\mathbf{g}}$$

$$\delta\dot{\mathbf{v}} = \mathbf{C}_b^i\left(\hat{\mathbf{q}}\right)\left[\boldsymbol{\alpha}\times\right]\hat{\mathbf{f}}^b + \mathbf{C}\left(\hat{\mathbf{q}}\right)_b^i\delta\mathbf{f} + \mathbf{C}\left(\hat{\mathbf{q}}\right)_b^i\left[\boldsymbol{\alpha}\times\right]\delta\mathbf{f} + \delta\mathbf{g} \tag{42}$$

Due to the small angle error assumption, it is assumed that

$$\mathbf{C}\left(\hat{\mathbf{q}}\right)_b^i\left[\mathbf{I} + \left[\boldsymbol{\alpha}\times\right]\right]\delta\mathbf{f} \cong \mathbf{C}\left(\hat{\mathbf{q}}\right)_b^i\delta\mathbf{f}\,. \tag{43}$$

Also, it is assumed for this application that the gravity error $\delta\mathbf{g}$ is negligible. Therefore, equation 42 can be expressed as a linear system with respect to the error states:

$$\delta\dot{\mathbf{v}} = -\mathbf{C}_b^i\left(\hat{\mathbf{q}}\right)\left[\hat{\mathbf{f}}^b\times\right]\boldsymbol{\alpha} + \mathbf{C}\left(\hat{\mathbf{q}}\right)_b^i\delta\mathbf{f}\,. \tag{44}$$

The position error state dynamics are:

$$\delta\dot{\mathbf{r}} = \dot{\mathbf{r}} - \dot{\hat{\mathbf{r}}} = \mathbf{v} - \hat{\mathbf{v}} = \delta\mathbf{v}\,. \tag{45}$$

The angular rate and specific force error states $\delta\boldsymbol{\omega}$ and $\delta\mathbf{f}$ are parametric, and can be used to model sensor calibration or noise terms. The angular rate error model that we will use here is a bias plus noise model:

$$\boldsymbol{\omega} = \hat{\boldsymbol{\omega}} - \boldsymbol{\beta}_\omega - \boldsymbol{\eta}_\omega \tag{46}$$

and

$$\dot{\boldsymbol{\beta}}_\omega = \boldsymbol{\upsilon}_\omega\,, \tag{47}$$

where $\boldsymbol{\eta}_\omega$ and $\boldsymbol{\upsilon}_\omega$ are white random processes with cross-correlations:

$$E\left[\boldsymbol{\eta}_\omega\left(t\right)\boldsymbol{\eta}_\omega^T\left(\tau\right)\right] = diag\left(\boldsymbol{\sigma}_\omega^2\right)\delta\left(t-\tau\right). \tag{48}$$

$$E\left[\boldsymbol{\upsilon}_\omega\left(t\right)\boldsymbol{\upsilon}_\omega^T\left(\tau\right)\right] = diag\left(\boldsymbol{\sigma}_{\beta\omega}^2\right)\delta\left(t-\tau\right). \tag{49}$$

The angular rate error term from equations 23 and 46 becomes:

$$\delta\boldsymbol{\omega} = -\boldsymbol{\beta}_{\omega} - \boldsymbol{\eta}_{\omega}. \tag{50}$$

$$\delta\dot{\boldsymbol{\omega}} = -\boldsymbol{\upsilon}_{\omega}. \tag{51}$$

The specific force measurements from the accelerometers are treated in a similar manner:

$$\mathbf{f} = \hat{\mathbf{f}} - \boldsymbol{\beta}_f - \boldsymbol{\eta}_f, \tag{52}$$

$$\dot{\boldsymbol{\beta}}_f = \boldsymbol{\upsilon}_f, \tag{53}$$

and

$$\delta\mathbf{f} = -\boldsymbol{\beta}_f - \boldsymbol{\eta}_f, \tag{54}$$

where $\boldsymbol{\eta}_f$ and $\boldsymbol{\upsilon}_f$ are white random processes with cross-correlations:

$$E\left[\boldsymbol{\eta}_f(t)\boldsymbol{\eta}_f^T(\tau)\right] = diag\left(\boldsymbol{\sigma}_f^2\right)\delta(t-\tau). \tag{55}$$

$$E\left[\boldsymbol{\upsilon}_f(t)\boldsymbol{\upsilon}_f^T(\tau)\right] = diag\left(\boldsymbol{\sigma}_{\beta f}^2\right)\delta(t-\tau). \tag{56}$$

Magnetometers will be used as measurements in the eventual Kalman filter implementation. The magnetometer is not an inertial sensor, and its error dynamics are uncoupled with the inertial navigation states, but a bias term is still augmented to the total error state vector since some correlation is expected to be introduced by the Kalman filter measurement update equations. The magnetometer bias vector $\boldsymbol{\beta}_m$ here is treated as a slowly diverging random walk process driven by the noise process $\boldsymbol{\upsilon}_m$ with cross correlation:

$$E\left[\boldsymbol{\upsilon}_m(t)\boldsymbol{\upsilon}_m^T(\tau)\right] = diag\left(\boldsymbol{\sigma}_{\beta m}^2\right)\delta(t-\tau). \tag{57}$$

$$\dot{\boldsymbol{\beta}}_m = \boldsymbol{\upsilon}_m. \tag{58}$$

The total error state vector is now:

$$\delta\mathbf{x} = \left\{\boldsymbol{\alpha}^T \quad \delta\mathbf{v}^T \quad \delta\mathbf{r}^T \quad \boldsymbol{\beta}_{\omega}{}^T \quad \boldsymbol{\beta}_f{}^T \quad \boldsymbol{\beta}_m{}^T\right\}^T. \tag{59}$$

The error state dynamics can be written in state space form:

$$\begin{Bmatrix} \dot{\boldsymbol{\alpha}} \\ \delta\dot{\mathbf{v}} \\ \delta\dot{\mathbf{r}} \\ \dot{\boldsymbol{\beta}}_\omega \\ \dot{\boldsymbol{\beta}}_f \\ \dot{\boldsymbol{\beta}}_m \end{Bmatrix} = \begin{bmatrix} -\left[\hat{\boldsymbol{\omega}}\times\right] & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ -\mathbf{C}_b^i\left(\hat{\mathbf{q}}\right)\left[\hat{\mathbf{f}}^b\times\right] & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{C}_b^i\left(\hat{\mathbf{q}}\right) & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix} \begin{Bmatrix} \boldsymbol{\alpha} \\ \delta\mathbf{v} \\ \delta\mathbf{r} \\ \boldsymbol{\beta}_\omega \\ \boldsymbol{\beta}_f \\ \boldsymbol{\beta}_m \end{Bmatrix} + \begin{Bmatrix} -\boldsymbol{\eta}_\omega \\ -\mathbf{C}_b^i\left(\hat{\mathbf{q}}\right)\boldsymbol{\eta}_f \\ \mathbf{0}_{3\times1} \\ \boldsymbol{\upsilon}_\omega \\ \boldsymbol{\upsilon}_f \\ \boldsymbol{\upsilon}_m \end{Bmatrix}. \qquad (60)$$

This 18-error-state, state-space model will be used for Kalman filter implementation in this work. An important result is that the model is linear and time-varying with respect to the error states. The advantage of this is that when the Kalman filter is implemented in realtime, the state transition matrix can be constructed without knowing the error states at the last measurement time from the information available from the IMU integration. Another advantage of this form is that there are no trigonometric operations involved with its computation. The third advantage is that the model is truly a stochastic system that is driven by white noise sequences that have spectral densities that are either measureable or easily approximated.

5. Process Noise Measurement

From equation 60, the following matrices are defined:

$$\mathbf{F}\left(\hat{\mathbf{q}},\hat{\boldsymbol{\omega}},\hat{\mathbf{f}}\right) = \begin{bmatrix} -\left[\hat{\boldsymbol{\omega}}\times\right] & \mathbf{0} & \mathbf{0} & -\mathbf{I}_{3\times3} & \mathbf{0} & \mathbf{0} \\ -\mathbf{C}_b^i\left(\hat{\mathbf{q}}\right)\left[\hat{\mathbf{f}}^b\times\right] & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{C}_b^i\left(\hat{\mathbf{q}}\right) & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{3\times3} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}. \qquad (61)$$

$$\mathbf{G}\left(\hat{\mathbf{q}}\right) = \begin{bmatrix} diag\left(-\boldsymbol{\sigma}_\omega\right) & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{C}_b^i\left(\hat{\mathbf{q}}\right)diag\left(\boldsymbol{\sigma}_f\right) & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & diag\left(\boldsymbol{\sigma}_{\beta\omega}\right) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & diag\left(\boldsymbol{\sigma}_{\beta f}\right) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & diag\left(\boldsymbol{\sigma}_{\beta m}\right) \end{bmatrix}. \qquad (62)$$

Using these definitions, the error state dynamics can be represented by the following matrix equation:

$$\delta\dot{\mathbf{x}} = \mathbf{F}\delta\mathbf{x} + \mathbf{G}\mathbf{w}\,. \tag{63}$$

The vector $\mathbf{w}$ is a white noise sequence with unity variance and zero mean. The matrix $\mathbf{G}$ transforms the white noise sequence into the disturbance vector. Therefore, the uncertainty of the disturbance vector can be represented by:

$$E\left[\left(\mathbf{G}\mathbf{w}(t)\right)\left(\mathbf{G}\mathbf{w}(\tau)\right)^T\right] = \mathbf{Q}_c\delta(t-\tau), \tag{64}$$

which simplifies to:

$$\mathbf{Q}_c\delta(t-\tau) = \mathbf{G}\mathbf{G}^T\delta(t-\tau) = \begin{bmatrix} diag\left(\boldsymbol{\sigma}_\omega^2\right) & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & diag\left(\boldsymbol{\sigma}_f^2\right) & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & diag\left(\boldsymbol{\sigma}_{\beta\omega}^2\right) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & diag\left(\boldsymbol{\sigma}_{\beta f}^2\right) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & diag\left(\boldsymbol{\sigma}_{\beta m}^2\right) \end{bmatrix}\delta(t-\tau)\,. \tag{65}$$

Representing the process noise as a continuous noise process simplifies filter tuning by making it sample time agnostic. However, discrete time integration leads to different variance growth than continuous time integration. Consider the integration of a scalar random signal. The integration of a continuous white noise signal with spectral amplitude σ_c^2 has the transfer function:

$$\frac{y(s)}{u(s)} = G(s) = \frac{\sigma_c}{s}\,. \tag{66}$$

Therefore, the variance of the output y at some time t is given by (*17*):

$$E[y^2(t)] = \sigma_c^2\int_0^t\int_0^t \delta(u-v)\,du\,dv = \sigma_c^2 t\,. \tag{67}$$

Consider the integration of a discrete white noise sequence $x_k \sim N(0,\sigma_d)$. If this sequence is integrated with rectangular integration, the output is given by:

$$y_k = \Delta t(x_k + x_{k-1} ... + x_0)\,. \tag{68}$$

The output variance is given by:

$$\begin{aligned} E[y_k^2] &= E[\Delta t[x_k + x_{k-1} ... + x_0]\Delta t[x_k + x_{k-1} ... + x_0]^T] \\ E[y_k^2] &= \Delta t^2 (E[x_k^2] + E[x_{k-1}^2] + ... E[x_o^2]) \\ E[y_k^2] &= \Delta t^2 k \sigma_d^2 = t \Delta t \sigma_d^2 \end{aligned} \quad . \qquad (69)$$

The standard deviation of the discrete white noise sequence is easily estimated from a short sample of a sensor's null output, and from it the spectral amplitude can be calculated for an equivalent continuous noise process. Therefore, in this work, the noise amplitudes for the angular rate sensors and accelerometers, $\boldsymbol{\sigma}_\omega^2$ and $\boldsymbol{\sigma}_f^2$, respectively, are calculated as a function of the estimated discrete time variance and the integration sample time:

$$\sigma_c = \hat{\sigma}_d \sqrt{\Delta t} \, . \qquad (70)$$

The drift-rate terms for the sensor biases are not analytically measured or simulated here. Allan variance analysis reveals the existence of flicker noise, but its measurement, simulation, and effect on a guided projectile's navigation solution are left as the subject of future work. Therefore, while the sensor noise parameters are measured, the bias drift parameters $\boldsymbol{\sigma}_{\beta\omega}$, $\boldsymbol{\sigma}_{\beta f}$, and $\boldsymbol{\sigma}_{\beta m}$ are chosen as tuning parameters.

6. Projectile Measurement Models

6.1 Vector Measurements

Because of the normal operating conditions for guided projectiles, there are only a few different kinds of measurements available. The most common are vector measurements in which a vector's coordinates are either known a-priori or measured in both the inertial and body-fixed frames. Since magnetometers are probably the most common vector measurement used by the projectile community, the Earth's magnetic field vector $\mathbf{m}$ will be used to illustrate the mapping of the error states to a vector's measurement residuals. Assuming the scale factor and misalignment terms are known, the magnetic field vector as measured by the magnetometer is given by:

$$\tilde{\mathbf{m}}^b = \mathbf{C}_i^b(\mathbf{q})\mathbf{m}^i + \boldsymbol{\beta}_m + \boldsymbol{\eta}_m \, . \qquad (71)$$

That is, the measured $\tilde{\mathbf{m}}^b$ is the known vector in inertial coordinates $\mathbf{m}^i$ transformed into the body coordinates, plus a body-fixed bias vector $\boldsymbol{\beta}_m$ and an additive zero-mean white noise term $\boldsymbol{\eta}_m$. The residual $\delta\mathbf{m}$ is formed by subtracting the predicted measurement $\hat{\tilde{\mathbf{m}}}^b$ from the actual measurement.

The predicted measurement is a function of the current estimated quaternion:

$$\hat{\tilde{\mathbf{m}}}^b = \mathbf{C}_i^b\left(\hat{\mathbf{q}}\right)\mathbf{m}^i . \tag{72}$$

Equation 37 can be substituted into equation 71 to get:

$$\tilde{\mathbf{m}}^b \cong \left[\mathbf{I} - \left[\boldsymbol{\alpha}\times\right]\right]\mathbf{C}_i^b\left(\hat{\mathbf{q}}\right)\mathbf{m}^i + \boldsymbol{\beta}_m + \boldsymbol{\eta}_m \tag{73}$$

The measurement residual is then:

$$\tilde{\mathbf{m}}^b - \hat{\tilde{\mathbf{m}}}^b = \left[\mathbf{I} - \left[\boldsymbol{\alpha}\times\right]\right]\mathbf{C}_i^b\left(\hat{\mathbf{q}}\right)\mathbf{m}^i + \boldsymbol{\beta}_m + \boldsymbol{\eta}_m - \mathbf{C}_i^b\left(\hat{\mathbf{q}}\right)\mathbf{m}^i , \tag{74}$$

which simplifies to a linear mapping of the error states:

$$\delta\tilde{\mathbf{m}}^b = \left[\left[\left(\mathbf{C}_i^b\left(\hat{\mathbf{q}}\right)\mathbf{m}^i\right)\times\right] \quad \mathbf{I}_{3\times3}\right]\begin{Bmatrix}\boldsymbol{\alpha}\\ \boldsymbol{\beta}_m\end{Bmatrix} + \boldsymbol{\eta}_m . \tag{75}$$

Another all-weather vector measurement is the projectile’s velocity vector in inertial coordinates $\tilde{\mathbf{v}}^i$. This can usually be obtained from an onboard GPS unit or from a reference trajectory or dynamic model depending on the mission specifications. Since most projectiles fly at a fairly low angle of attack, the body fixed coordinates are assumed to be:

$$\tilde{\mathbf{v}}^b = \left\{\left\|\tilde{\mathbf{v}}^b\right\| \quad 0 \quad 0\right\}^T . \tag{76}$$

The noise in the velocity vector is usually more naturally expressed in inertial coordinates. The measurement residual is given by:

$$\delta\tilde{\mathbf{v}}^i = \tilde{\mathbf{v}}^i - \mathbf{C}_b^i\left(\hat{\mathbf{q}}\right)\tilde{\mathbf{v}}^b . \tag{77}$$

The measured velocity in inertial coordinates can be rewritten in terms of the error states and the noise vector $\boldsymbol{\eta}_v$:

$$\tilde{\mathbf{v}}^i \cong \mathbf{C}_b^i\left(\hat{\mathbf{q}}\right)\left[\mathbf{I} + \left[\boldsymbol{\alpha}\times\right]\right]\tilde{\mathbf{v}}^b + \boldsymbol{\eta}_v . \tag{78}$$

This leads to the following linear mapping of the error states to the velocity vector residual:

$$\delta\tilde{\mathbf{v}}^i = -\mathbf{C}_b^i\left(\hat{\mathbf{q}}\right)\left[\tilde{\mathbf{v}}^b\times\right]\boldsymbol{\alpha} + \boldsymbol{\eta}_v \tag{79}$$

6.2 Angle Measurements

Some measurement sources provide direct measurements of Euler angles. For example, a constellation of thermopile sensors with the appropriate signal processing provides a direct measurement of the roll angle ϕ from a ZYX Euler angle rotation sequence, except in specific weather conditions (*2*, *3*). Certain computer vision algorithms can use horizon detection to measure both the pitch angle θ as well as ϕ (*20*). Arguably, it might not make sense to even be using a quaternion-based attitude representation in this situation, but if quaternions are more suited for the mission requirements it is still simple to use Euler angle measurements. In this case, one would compare the measured Euler angles to the Euler angles computed from the current quaternion prediction to get Euler angle error residuals. That is, for a typical ZYX Euler angle rotation sequence:

$$\delta\tilde{\phi} = \tilde{\phi} - \hat{\tilde{\phi}} + \eta_{\phi}, \tag{80}$$

$$\delta\tilde{\theta} = \tilde{\theta} - \hat{\tilde{\theta}} + \eta_{\theta}, \tag{81}$$

and

$$\delta\tilde{\psi} = \tilde{\psi} - \hat{\tilde{\psi}} + \eta_{\psi}, \tag{82}$$

where:

$$\hat{\tilde{\phi}} = \text{atan2}\left(2\left(\hat{q}_3\hat{q}_4 + \hat{q}_1\hat{q}_2\right), \hat{q}_1^2 - \hat{q}_2^2 - \hat{q}_3^2 + \hat{q}_4^2\right). \tag{83}$$

$$\hat{\tilde{\theta}} = \left(-\sin^{-1}\left(2\left(\hat{q}_2\hat{q}_4 - \hat{q}_1\hat{q}_3\right)\right)\right). \tag{84}$$

$$\hat{\tilde{\psi}} = \text{atan2}\left(2\left(\hat{q}_2\hat{q}_3 + \hat{q}_1\hat{q}_4\right), \hat{q}_1^2 + \hat{q}_2^2 - \hat{q}_3^2 - \hat{q}_4^2\right). \tag{85}$$

A quaternion Kalman filter estimates small rotation angles however, so a mapping must be used between the Euler angle residuals, and small rotation angle error states. Assuming that both Euler angle residuals and error states are small angles, the mapping between the two is identical to the mapping between Euler angle time derivatives and the body-fixed angular rate vector:

$$\begin{Bmatrix} \delta\tilde{\phi} \\ \delta\tilde{\theta} \\ \delta\tilde{\psi} \end{Bmatrix} = \begin{bmatrix} 1 & \sin\left(\hat{\tilde{\phi}}\right)\tan\left(\hat{\tilde{\theta}}\right) & \cos\left(\hat{\tilde{\phi}}\right)\tan\left(\hat{\tilde{\theta}}\right) \\ 0 & \cos\left(\hat{\tilde{\phi}}\right) & -\sin\left(\hat{\tilde{\phi}}\right) \\ 0 & \sin\left(\hat{\tilde{\phi}}\right)/\cos\left(\hat{\tilde{\theta}}\right) & \cos\left(\hat{\tilde{\phi}}\right)/\cos\left(\hat{\tilde{\theta}}\right) \end{bmatrix} \boldsymbol{\alpha}. \tag{86}$$

When implementing Euler angle measurements, it is very important to make sure that the measured and predicted quantities are in the same quadrant.

6.3 Position and Velocity Measurements

The availability and source of actual position measurements is highly subjective to mission requirements. The mapping from the error states to direct position or velocity residuals is simple:

$$\tilde{\mathbf{r}} - \hat{\tilde{\mathbf{r}}} = \left[\mathbf{I}\right]\delta\mathbf{r} + \boldsymbol{\eta}_r. \tag{87}$$

$$\tilde{\mathbf{v}} - \hat{\tilde{\mathbf{v}}} = \left[\mathbf{I}\right]\delta\mathbf{v} + \boldsymbol{\eta}_v \,. \tag{88}$$

In reality, the measurements are usually more complicated. If the user has access to the direct GPS pseudo-range and carrier-frequency measurements, it would be more accurate to use the well-known GPS measurement models that include user clock bias. If the user does not have access to the direct receiver outputs but instead is provided with the receiver's calculated position and velocity estimate, it may be more accurate to measure the errors to construct an auto-regressive or Gauss-Markov error model instead of assuming white noise. In the case of a reference trajectory, it may be beneficial to treat the dynamic model as a random process so that the reference trajectory "measurements" get weighed less and less as the trajectory progresses in time. A simple way to do this is to model the errors in reference trajectory states as three coupled position velocity states driven by random noise:

$$\begin{Bmatrix} \delta\dot{\tilde{r}}_{x-z} \\ \delta\dot{\tilde{v}}_{x-z} \end{Bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} \delta\tilde{r}_{x-z} \\ \delta\tilde{v}_{x-z} \end{Bmatrix} + \boldsymbol{\eta}_{ref} \,. \tag{89}$$

Deviations from the reference trajectory due to unmodeled disturbances such as wind would be accounted for in the noise term $\boldsymbol{\eta}_{ref}$ which is parameterized by the tuning variable q_{ref}:

$$E\left[\boldsymbol{\eta}_{ref}\left(t\right)\boldsymbol{\eta}_{ref}^{T}\left(\tau\right)\right] = \begin{bmatrix} 0 & 0 \\ 0 & q_{ref}^{2} \end{bmatrix} \delta\left(t-\tau\right). \tag{90}$$

The measurement covariance at some time later can be calculated from an initial value $\mathbf{R}_{ref\,0}$ and the tuning variable:

$$E\left[\begin{Bmatrix} \delta\tilde{r}\left(t\right)_{x-z} \\ \delta\tilde{v}\left(t\right)_{x-z} \end{Bmatrix} \begin{Bmatrix} \delta\tilde{r}\left(t\right)_{x-z} \\ \delta\tilde{v}\left(t\right)_{x-z} \end{Bmatrix}^{T}\right] = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix} \mathbf{R}_{ref\,0} \begin{bmatrix} 1 & 0 \\ t & 1 \end{bmatrix} + \begin{bmatrix} t^{3}/3 & t^{2}/2 \\ t^{2}/2 & t \end{bmatrix} q_{ref}^{2} \,. \tag{91}$$

This measurement covariance matrix would then be used in the EKF when calculating the gains for the position and velocity measurements.

7. Kalman Filtering Implementation

The Kalman filter is eventually intended to be run on an embedded processor. In this work, this is accomplished by discretizing the error state dynamics so that the Kalman filtering equations can be updated at a lower frequency than the integrated states. The integration of the gyroscopes and accelerometers to acquire quaternion, velocity, and position will be updated at a faster rate than the EKF since the integration should require significantly less computing power and delayed feedback can degrade the performance of flight controllers. Assuming some type of multitasking is used, the Kalman filter equations are run at a slower rate with a lower priority as shown in table 1.

Table 1. Kalman filter multitasking schedule.

IMU Integration Time i =		**EKF Update Time j =**	
0	Start at $\hat{\mathbf{x}}_0^-$	**1**	Measurements $\mathbf{y}_{j=0}$ Kalman Filtering Calculations Updated $\hat{\mathbf{x}}_{j=0}^+$ known Predict $\hat{\mathbf{x}}_{j=1}^-$
1	Integrate $\hat{\mathbf{x}}^-$ from $\hat{\mathbf{x}}_{i=0}^-$		
2	Integrate $\hat{\mathbf{x}}^-$ from $\hat{\mathbf{x}}_{i=1}^-$		
3	Integrate $\hat{\mathbf{x}}^-$ from $\hat{\mathbf{x}}_{i=2}^-$		
4	Integrate $\hat{\mathbf{x}}^-$ from $\hat{\mathbf{x}}_{i=3}^-$		
5	Integrate $\hat{\mathbf{x}}^-$ from $\mathbf{x}_{j=1}^-$	**2**	Measurements $\mathbf{y}_{j=0}$ Kalman Filtering Calculations Updated $\hat{\mathbf{x}}_{j=1}^+$ known Predict $\hat{\mathbf{x}}_{j=2}^-$
6	Integrate $\hat{\mathbf{x}}^-$ from $\hat{\mathbf{x}}_{i=5}^-$		
7	Integrate $\hat{\mathbf{x}}^-$ from $\hat{\mathbf{x}}_{i=6}^-$		
8	Integrate $\hat{\mathbf{x}}^-$ from $\hat{\mathbf{x}}_{i=7}^-$		
9	Integrate $\hat{\mathbf{x}}^-$ from $\hat{\mathbf{x}}_{i=8}^-$		
—	Etc.	**—**	Etc.

The integration of the full states is accomplished using the equations in section 5 and the best known gyroscope, and accelerometer calibrations. The actual Kalman filtering calculations are performed to estimate the error states at the time of the measurements. Each Kalman filter iteration consists of the following five steps:

1. *Measurement/Storage*

Record the measurements and their covariance values (if time varying) from whatever sources are available. Record the value of integrated full states at this time, as they will be used to form the residuals.

2. *Covariance Prediction*

The full state prediction comes from the integration task. The a-priori error-state estimate is zero:

$$\delta\hat{\mathbf{x}}_j^- = \mathbf{0}\,. \tag{92}$$

An estimate of the a-priori error-state covariance matrix $\mathbf{P}_j^-$ must be predicted from the previous iteration's a-posteriori error-state covariance $\mathbf{P}_{j-1}^+$ and the error-state dynamics. The matrix $\mathbf{F}$ previously defined in equation 61 can be discretized to form a state transition matrix using the second-order approximation for the matrix exponential:

$$\mathbf{\Phi} = \mathbf{I} + \mathbf{F}\Delta t + \frac{1}{2}\mathbf{F}^2\Delta t^2\,. \tag{93}$$

The values that go into the $\mathbf{F}$ matrix are determined from the current a-priori quaternion estimate and IMU outputs, and the previous iteration's a-posteriori quaternion estimate and IMU outputs. The nonzero terms in $\mathbf{F}$ are therefore:

$$\begin{aligned}
&-\left[\hat{\boldsymbol{\omega}}\times\right] = -\left[\left(\hat{\boldsymbol{\omega}}_{j-1} + \hat{\boldsymbol{\omega}}_j\right)/2\times\right] \\
&-\mathbf{C}_b^i\left(\hat{\mathbf{q}}\right)\left[\hat{\mathbf{f}}^b\times\right] = \frac{-\mathbf{C}_b^i\left(\hat{\mathbf{q}}_j^-\right)\left[\hat{\mathbf{f}}_j^b\times\right] + -\mathbf{C}_b^i\left(\hat{\mathbf{q}}_{j-1}^+\right)\left[\hat{\mathbf{f}}_{j+1}^b\times\right]}{2}\,. \\
&-\mathbf{C}_b^i\left(\hat{\mathbf{q}}\right) = -\mathbf{C}_b^i\left(\bar{\mathbf{q}}_{j-1}^j\right)
\end{aligned} \tag{94}$$

Note that the term $\bar{\mathbf{q}}_{j-1}^j$ is the average quaternion determined from the algorithm in appendix B, as arithmetic averaging of two rotation transformation matrices violates the orthonormal constraints. The *a-priori* error state covariance is calculated as:

$$\mathbf{P}_j^- = E\left[\delta\mathbf{x}\delta\mathbf{x}^T\right] = \mathbf{\Phi}\mathbf{P}_{j-1}^+\mathbf{\Phi}^T + \mathbf{Q}_d\,. \tag{95}$$

The discrete time process noise covariance matrix $\mathbf{Q}_d$ is determined from:

$$\mathbf{Q}_d = \int_0^{\Delta t} e^{\mathbf{F}(t-\tau)}\mathbf{Q}_c e^{\mathbf{F}^T(t-\tau)}d\tau\,. \tag{96}$$

The previous equation can be simplified considerably by assuming that the $\mathbf{F}$ matrix is linearized about the conditions $\hat{\boldsymbol{\omega}} = \hat{\mathbf{f}} = 0$ and $\mathbf{C}_i^b\left(\hat{\mathbf{q}}\right) = \mathbf{I}$. This makes some sense intuitively, works in practice, and is used in reference 11, but an analytical reason why this works is left as a subject of future research. With this simplification and the definition of $\mathbf{Q}_c$ defined in equation 65 an analytical expression can be determined. With the notational shorthand $\boldsymbol{\Lambda}\left(\mathbf{v}\right)$ meaning a matrix with the elements of the vector $\mathbf{v}$ along the diagonal, the analytical expression is:

$$\mathbf{Q}_d = \begin{bmatrix} \boldsymbol{\Lambda}\left(\boldsymbol{\sigma}_\omega^2\right)\Delta t + \boldsymbol{\Lambda}\left(\boldsymbol{\sigma}_{\beta\omega}^2\right)\frac{\Delta t^3}{3} & \mathbf{0} & \mathbf{0} & -\boldsymbol{\Lambda}\left(\boldsymbol{\sigma}_{\beta\omega}^2\right)\frac{\Delta t^2}{2} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Lambda}\left(\boldsymbol{\sigma}_f^2\right)\Delta t + \boldsymbol{\Lambda}\left(\boldsymbol{\sigma}_{\beta f}^2\right)\frac{\Delta t^3}{3} & \boldsymbol{\Lambda}\left(\boldsymbol{\sigma}_{\beta f}^2\right)\frac{\Delta t^4}{8} + \boldsymbol{\Lambda}\left(\boldsymbol{\sigma}_f^2\right)\frac{\Delta t^2}{2} & \mathbf{0} & -\boldsymbol{\Lambda}\left(\boldsymbol{\sigma}_{\beta f}^2\right)\frac{\Delta t^2}{2} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Lambda}\left(\boldsymbol{\sigma}_f^2\right)\frac{\Delta t^2}{2} + \boldsymbol{\Lambda}\left(\boldsymbol{\sigma}_{\beta f}^2\right)\frac{\Delta t^4}{8} & \boldsymbol{\Lambda}\left(\boldsymbol{\sigma}_f^2\right)\frac{\Delta t^3}{3} + \boldsymbol{\Lambda}\left(\boldsymbol{\sigma}_{\beta f}^2\right)\frac{\Delta t^5}{20} & \mathbf{0} & -\boldsymbol{\Lambda}\left(\boldsymbol{\sigma}_{\beta f}^2\right)\frac{\Delta t^3}{6} & \mathbf{0} \\ -\boldsymbol{\Lambda}\left(\boldsymbol{\sigma}_{\beta\omega}^2\right)\frac{\Delta t^2}{2} & \mathbf{0} & \mathbf{0} & \boldsymbol{\Lambda}\left(\boldsymbol{\sigma}_{\beta\omega}^2\right)\frac{\Delta t^2}{2} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\boldsymbol{\Lambda}\left(\boldsymbol{\sigma}_{\beta f}^2\right)\frac{\Delta t^2}{2} & -\boldsymbol{\Lambda}\left(\boldsymbol{\sigma}_{\beta f}^2\right)\frac{\Delta t^3}{6} & \mathbf{0} & \boldsymbol{\Lambda}\left(\boldsymbol{\sigma}_{\beta f}^2\right)\Delta t & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{\Lambda}\left(\boldsymbol{\sigma}_{\beta m}^2\right)\Delta t \end{bmatrix}. \tag{97}$$

3. *Evaluate residual mappings*

All of the linear maps from the error states to the residuals in section 6 are functions of the current full state vector. In the case of linear Kalman filters, if all of the measurements are uncorrelated the measurements can be processed one at a time with identical results as processing them all at once (*17*). This adds convenience and reduces computation time since matrix inversion is avoided. To achieve the same effect with an EKF, the measurement maps must all be evaluated using the same a-priori full state estimate.

4. *Sequential Kalman filter innovation steps*

For each available measurement the following equations steps are taken. All of the residual maps in section 6 can be put into the form:

$$\delta y = \mathbf{h}\delta\mathbf{x} + \eta\,, \tag{98}$$

where δy is equal to the actual measurement minus the predicted measurement. The noise η will have some known covariance σ_y^2. The Kalman gain is then:

$$\mathbf{k} = \frac{\mathbf{P}_j^-\mathbf{h}^T}{\left(\mathbf{h}\mathbf{P}_j^-\mathbf{h}^T + \sigma_y^2\right)}\,. \tag{99}$$

The error state vector is updated with the Kalman gain:

$$\delta\hat{\mathbf{x}}_j^+ = \delta\hat{\mathbf{x}}_j^- + \mathbf{k}\delta y\,. \tag{100}$$

The error state covariance matrix is updated next:

$$\mathbf{P}_j^+ = \left(\mathbf{I} - \mathbf{kh}\right)\mathbf{P}_j^- . \tag{101}$$

If there are more measurements to be processed, then the a-priori error states and covariance matrix are overwritten as $\delta\hat{\mathbf{x}}_j^- = \delta\hat{\mathbf{x}}_j^+$ and $\mathbf{P}_j^- = \mathbf{P}_j^+$ before the innovation steps are repeated with the next measurement.

5. *Update the full states and calibration parameters*

Using the definitions for each error state as defined in section IV the full state estimate can be updated with the error state estimate. For the quaternion, this amounts to:

$$\hat{\mathbf{q}}_j^+ = \hat{\mathbf{q}}_j^- \otimes \begin{Bmatrix} 1 \\ \boldsymbol{\alpha}/2 \end{Bmatrix} . \tag{102}$$

The position and velocity estimates are updated as:

$$\hat{\mathbf{r}}_j^+ = \hat{\mathbf{r}}_j^- + \delta\mathbf{r} . \tag{103}$$

$$\hat{\mathbf{v}}_j^+ = \hat{\mathbf{v}}_j^- + \delta\mathbf{v} . \tag{104}$$

The bias estimates for the magnetometer, gyroscopes, and accelerometers are used to correct the calibrations for each sensor, and then set to zero. At this point, the full state estimates are updated at time j, which will have been some distance in the past depending on the speed of the processor. A current full state estimate can be obtained by using the IMU integration equations in section 3 and the outputs of the gyroscopes and accelerometers at time j and the current time. Note that when applying equation 70 to calculate the continuous random walk coefficients, better performance is usually obtained by using the time difference between the current time and time j when this time step is large instead of the integration time step.

8. Simulation Results

To validate some of the models and algorithms presented here, simulations were constructed in a Matlab/Simulink environment that included full six-degree-of-freedom (6-DOF) motion of the projectile as well as the state estimation, guidance, and control algorithms. The simulations are not to provide performance metrics for any specific system, but to demonstrate the basic functionality of the algorithms discussed in this report and discuss some of the problems with guided projectile applications.

8.1 Direct-Fire Application

The first system simulation is a low QE guide to hit mission against a moving target. The projectile used was the U.S. Army/Navy standard finner (*21–24*), which is a fin-stabilized nonspinning round, modified with four canards as shown in figure 3.

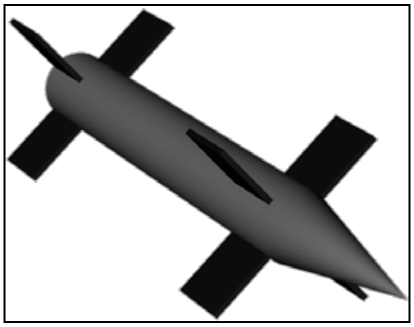

Figure 3. U.S. Army/Navy standard finner.

The projectile uses proportional navigation as the guidance law to engage the moving target. The autopilot is a state-feedback controller designed using optimal control techniques (*25*). The initial conditions for the simulation are shown in table 2.

Table 2. Direct-fire initial conditions.

Quantity	Value	Units
Muzzle velocity	200	m/s
Gun elevation	8	deg
Projectile initial position	$\{0\ \ 0\ \ 0\}^T$	m
Target initial position	$\{1000\ \ 0\ 0\}^T$	m
Target velocity	$\{0\ \ 10\ \ 0\}^T$	m/s
Projectile initial angular velocity	$\{2\ \ 2\ \ 2\}^T$	rad/s

The state estimator is only an attitude estimator in this case, as no actual position or velocity measurements are available. The gyros are corrupted by white noise with the same parameters measured from some modified ADXRS300 angular rate sensors, although no flicker or rate-ramp was modeled. Magnetometer measurements are present with the same noise measured from an HMC1043 solid-state magnetometer. It is assumed that relatively large initial biases might be present in the magnetometer due to body-fixed static magnetic fields generated by the projectile itself. The effects of motors and eddy current effects are neglected. No misalignment, scale factor, non-orthogonality, or g-sensitivity errors are modeled for either sensor. In addition to magnetometers, it is assumed that a computer vision horizon detection algorithm produces roll and pitch estimates at 30 Hz. It is also assumed that the computer-vision and EKF processing takes 80% of the update interval (26.7 ms) to complete. It turns out the yaw angle ψ is unobservable with this configuration (it is indistinguishable from the x-axis magnetometer bias). To remedy this, a heuristic yaw angle measurement of 0° is used for the first 0.5 s to enable the sensor biases to converge. The EKF parameters are shown in table 3, and errors are illustrated in figures 4–6.

Table 3. Direct-fire EKF parameters.

Quantity	Value	Units
Initial attitude error covariance $E\left[\boldsymbol{\alpha}\boldsymbol{\alpha}^T\right](t=0)$	diag{100 1 1}^{2}	deg^2
Initial gyro bias error covariance $E\left[\boldsymbol{\beta}_\omega\boldsymbol{\beta}_\omega{}^T\right](t=0)$	diag{10 10 10}^{2}	(deg/s)^2
Initial magnetometer bias error covariance $E\left[\boldsymbol{\beta}_m\boldsymbol{\beta}_m{}^T\right](t=0)$	diag{1 1 1}^{2}	Gauss^2
Gyro bias drift covariance $diag\left(\boldsymbol{\sigma}^2_{\beta\omega}\right)$	diag{1 1 1}x1e-3	(rad/s/s)^2
Magnetometer bias drift covariance $diag\left(\boldsymbol{\sigma}^2_{\beta m}\right)$	diag{1 1 1}x1e-3	(Gauss/s)^2
Gyroscope noise std. $\boldsymbol{\sigma}_{\omega d}$	{ 2.0626 2.0626 2.0626 }^{T}	deg/s
Magnetometer noise std. $\boldsymbol{\sigma}_{md}$	{0.0022 0.0022 0.0022}^{T}	Gauss
Horizon roll/pitch measurement noise std.	1	Deg
Quaternion integration rate	500	Hz
Kalman filter update rate	30	Hz
Earth's magnetic field in gun-target line coordinates	{0.5 0 0}^{T}	Gauss

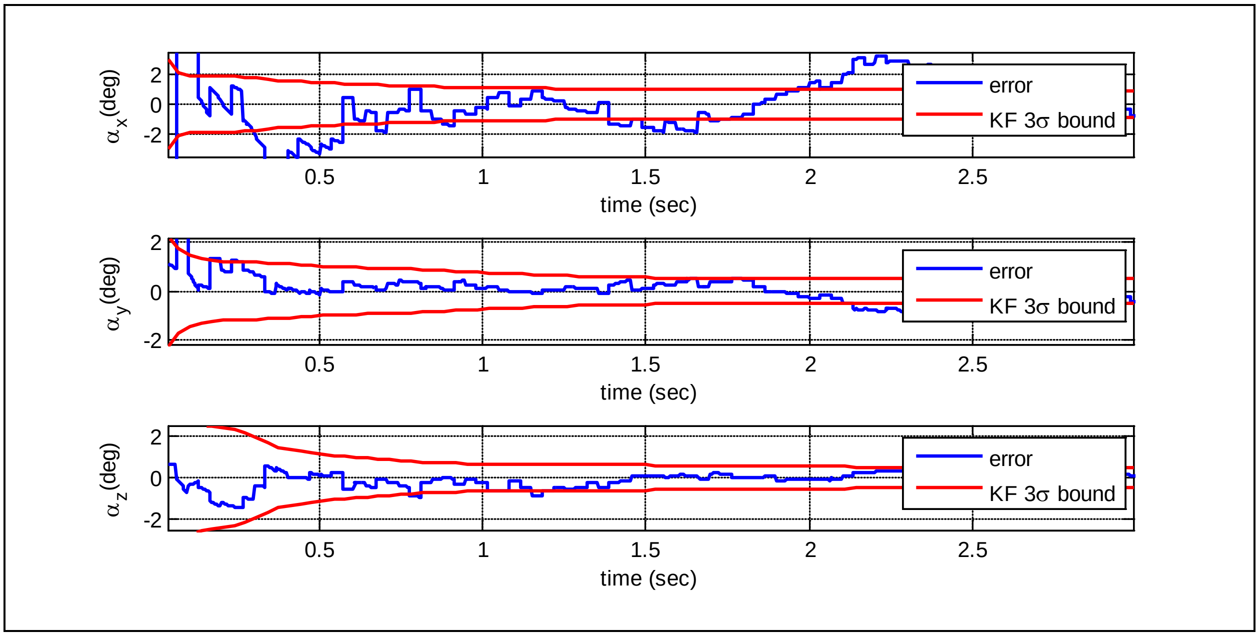

Figure 4. Direct-fire small-angle errors.

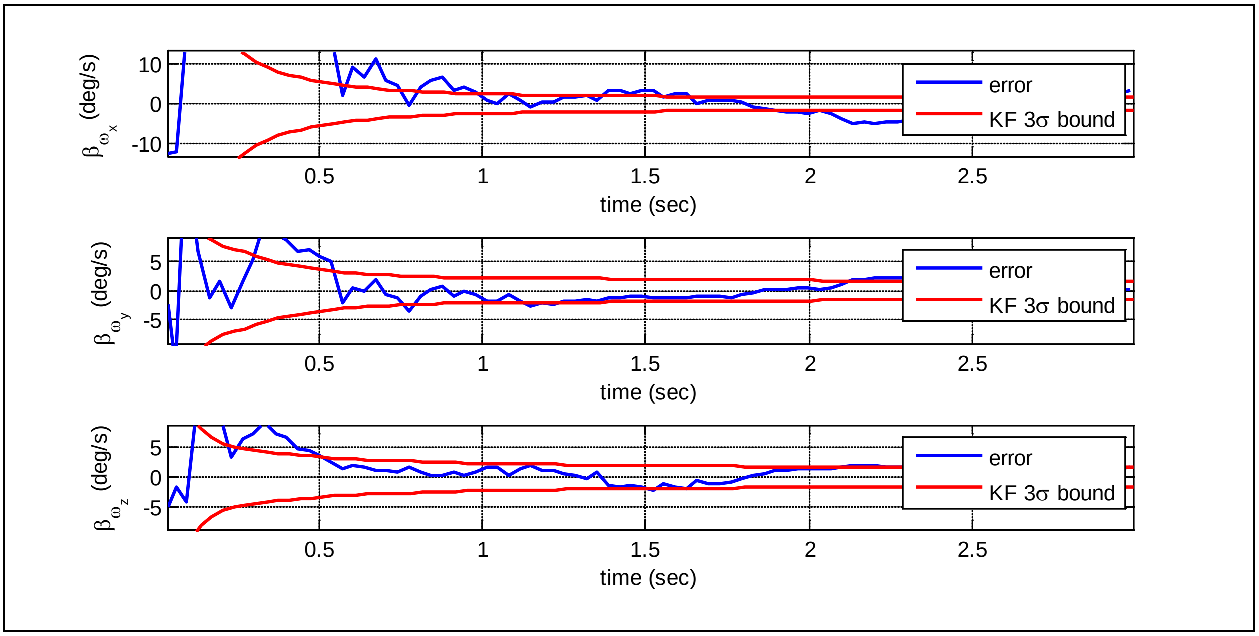

Figure 5. Direct-fire gyro-bias errors.

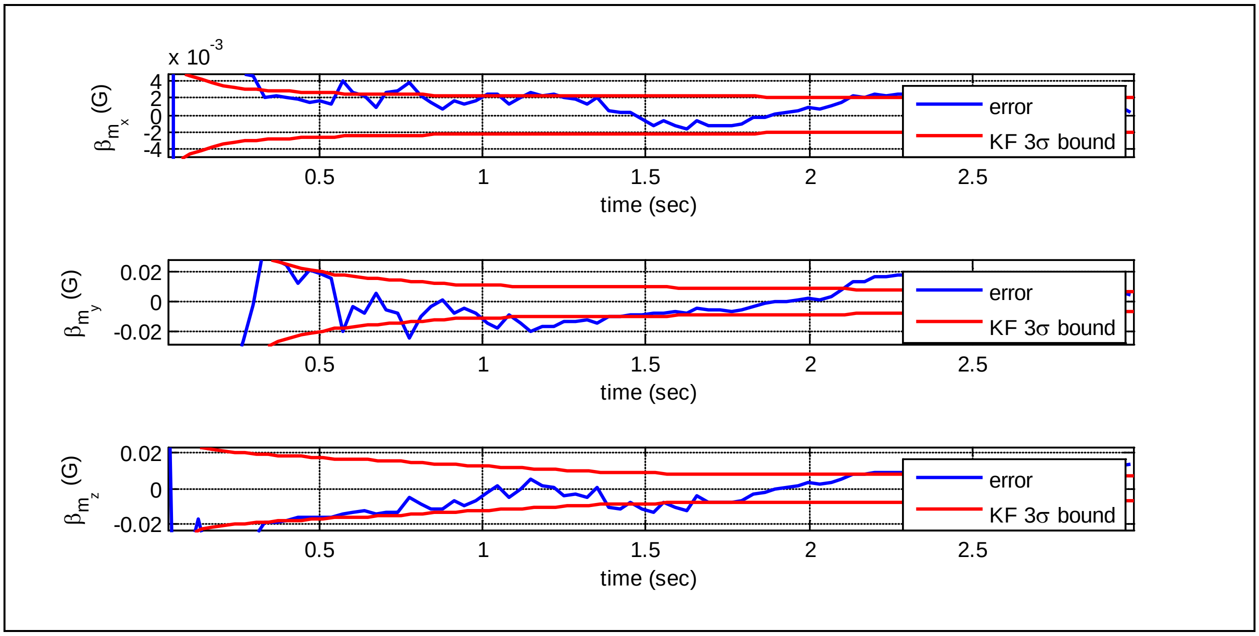

Figure 6. Direct-fire magnetometer-bias errors.

The filtered errors are not particularly impressive, but they might very well be usable for this kind of mission. The filter is slightly overconfident, which is evinced by relatively high amount of error that lies outside of the three standard deviation bound calculated from the estimated error covariance matrix. This is common in extended Kalman filters because covariance estimates are based on linearized nonlinear mapping functions.

8.2 Indirect-Fire Application

The second application is the navigation of an 81-mm mortar equipped with four independently controlled canards that deploy near apogee. The projectile is assumed to be equipped ADXRS300 angular rate sensors, a HMC1043 solid-state magnetometer, and an ADXL278 accelerometer triad, as well as an idealized position and velocity measurement for 30 s. The position and velocity measurement is just corrupted with additive white noise, while being available at 100 Hz. The initial conditions of the trajectory are listed in table 4, and plot of the trajectory is shown below in figure 7. Before the canards deploy, the mortar flies along a ballistic trajectory, which entails spinning up to 20 Hz. The fact that real ADXRS300 sensors would clip at this spin rate is ignored in this idealized example.

Table 4. Indirect-fire initial conditions.

Quantity	**Value**	**Units**
Muzzle velocity	274	m/s
Gun elevation	45	deg
Projectile initial position	$\{0\ \ 0\ \ 0\}^T$	m
Target initial position	$\{4900\ \ 100\ 0\}^T$	m
Target velocity	$\{0\ \ 0\ \ 0\}^T$	m/s
Projectile initial angular velocity	$\{0\ \ 4\ \ 4\}^T$	rad/s

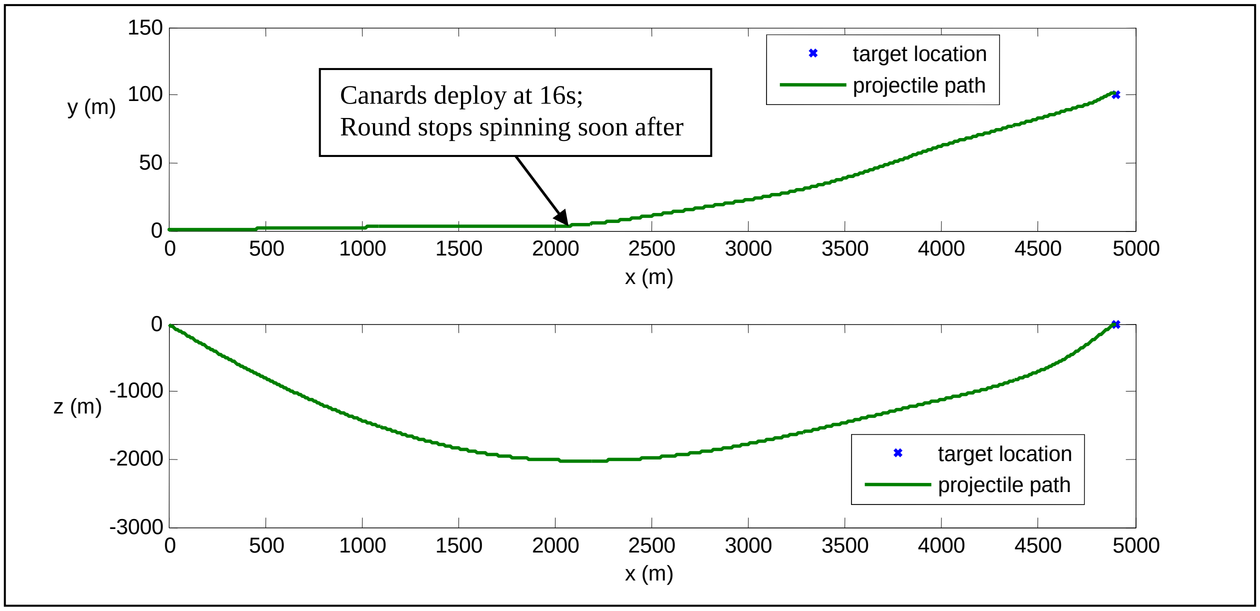

Figure 7. Indirect-fire trajectory.

The parameters for the sensors and Kalman filter are shown in table 5. A third idealization is that the accelerometers are located at the projectile center of gravity. The error results are shown in figures 8–13.

Table 5. Indirect-fire EKF parameters.

Quantity	**Value**	**Units**
Initial attitude error covariance $E\left[\boldsymbol{\alpha}\boldsymbol{\alpha}^T\right]\left(t=0\right)$	diag{100 1 1}^{2}	deg^2
Initial gyro bias error covariance $E\left[\boldsymbol{\beta}_\omega\boldsymbol{\beta}_\omega{}^T\right]\left(t=0\right)$	diag{10 10 10}^{2}	(deg/s)^2
Initial accelerometer bias error covariance $E\left[\boldsymbol{\beta}_f\boldsymbol{\beta}_f{}^T\right]\left(t=0\right)$	diag{10 10 10}^{2}	(m/s^2)^2
Initial magnetometer bias error covariance $E\left[\boldsymbol{\beta}_m\boldsymbol{\beta}_m{}^T\right]\left(t=0\right)$	diag{1 1 1}^{2}	Gauss^2
Gyro bias drift covariance $diag\left(\boldsymbol{\sigma}^2_{\beta\omega}\right)$	diag{1 1 1}x1e-3	(rad/s/s)^2
Accelerometer bias drift covariance $diag\left(\boldsymbol{\sigma}^2_{\beta f}\right)$	diag{1 1 1}x1e-3	(m/s^2/s)^2
Magnetometer bias drift covariance $diag\left(\boldsymbol{\sigma}^2_{\beta m}\right)$	diag{1 1 1}x1e-3	(Gauss/s)^2
Gyroscope noise std. $\boldsymbol{\sigma}_{\omega d}$	{ 2.0626 2.0626 2.0626 }^{T}	deg/s
Accelerometer noise std. $\boldsymbol{\sigma}_{fd}$	{ 0.3556 0.3556 0.3556 }^{T}	m/s^2
Magnetometer noise std. $\boldsymbol{\sigma}_{md}$	{0.0022 0.0022 0.0022}^{T}	Gauss
Heuristic yaw measurement noise std.	1	Deg
Heuristic yaw measurement duration	14	s
Position measurement noise std.	1	m
Position measurement duration	30	s
Velocity measurement noise std.	1	m/s
Velocity measurement duration	30	s
Velocity vector as an attitude measurement duration	14	s
Quaternion integration rate	500	Hz
Kalman filter update rate	100	Hz
Earth's magnetic field in gun-target line coordinates	{0.5 0 0}^{T}	Gauss

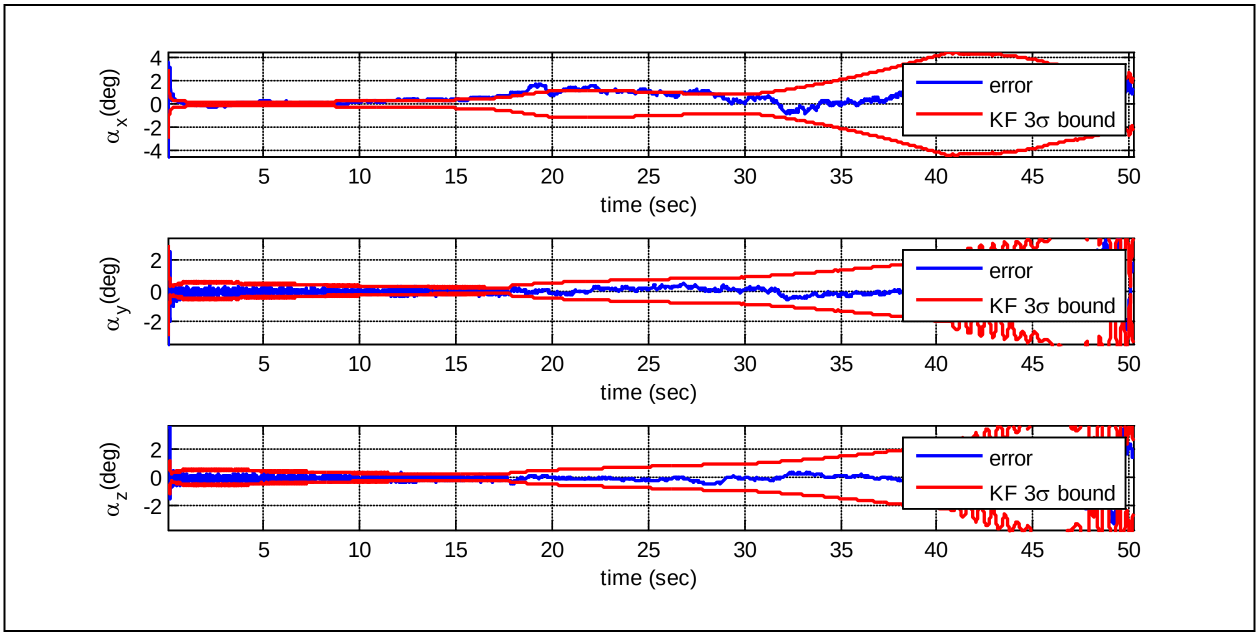

Figure 8. Indirect-fire angle errors.

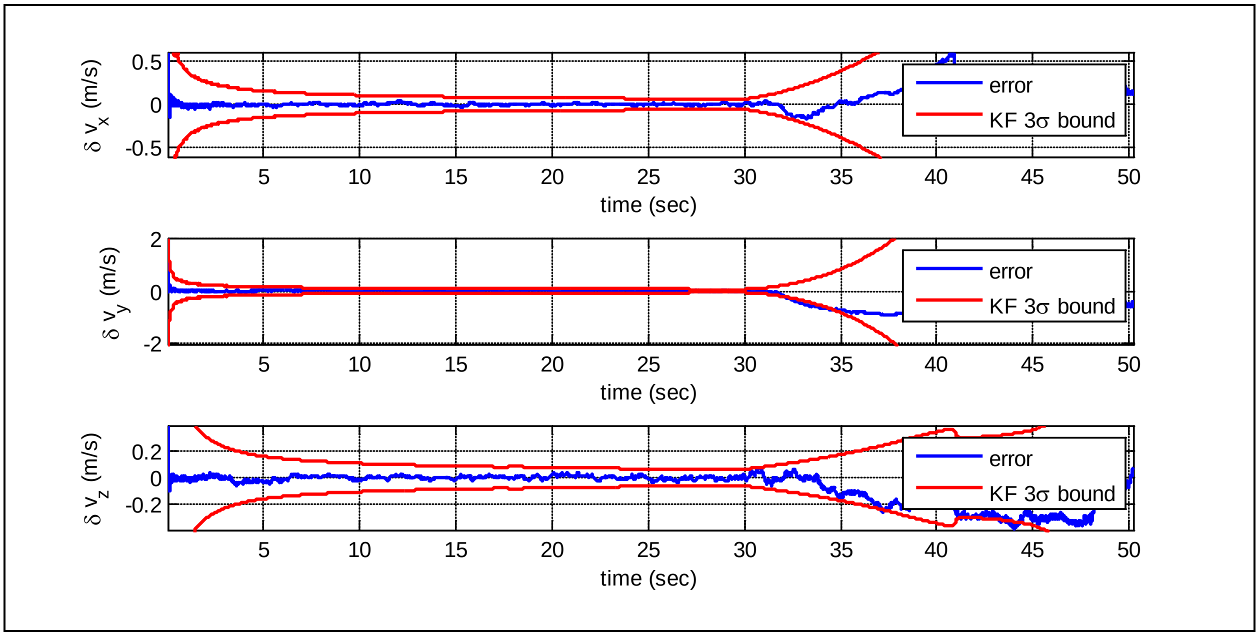

Figure 9. Indirect-fire velocity errors.

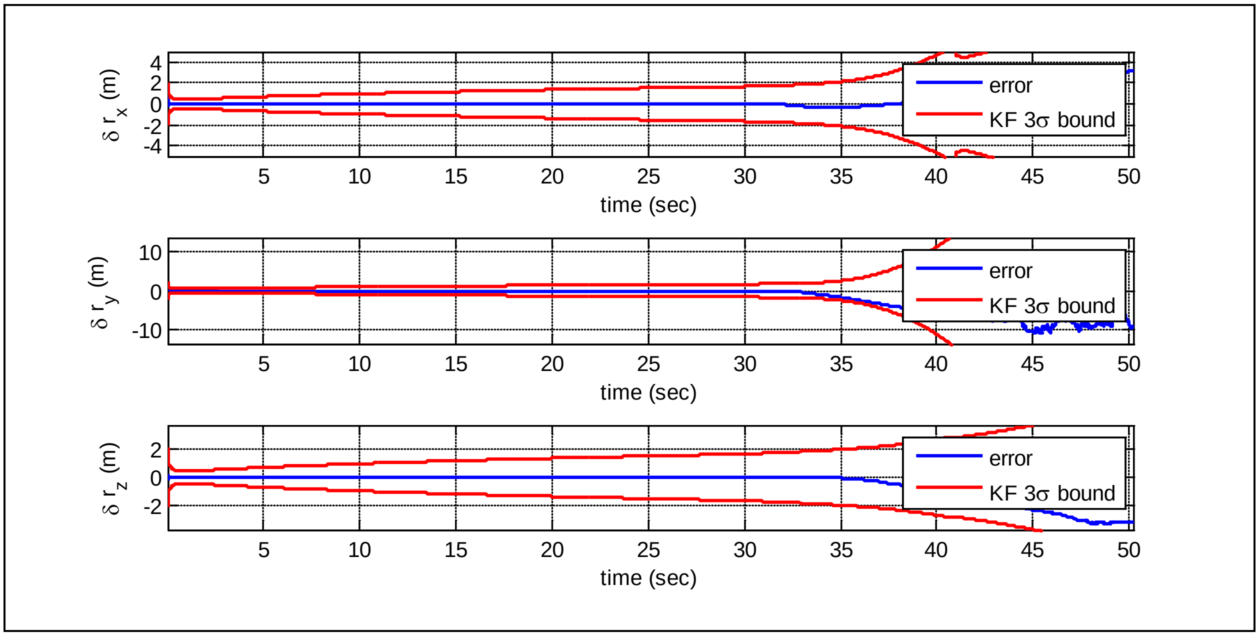

Figure 10. Indirect-fire position errors.

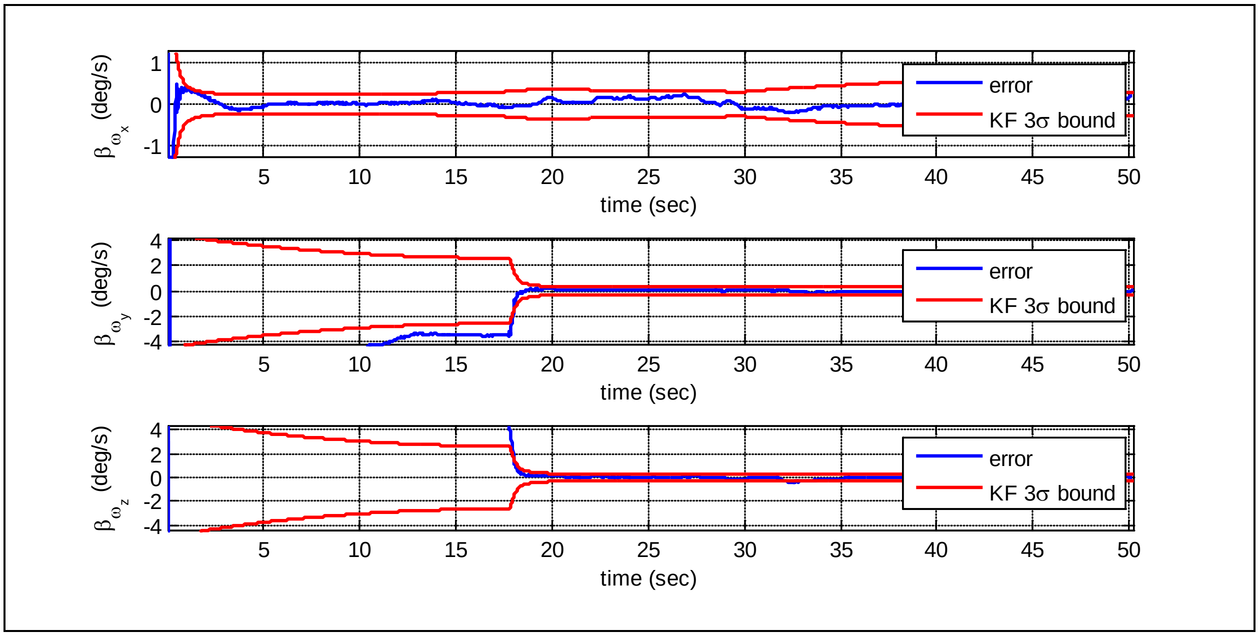

Figure 11. Indirect-fire gyro-bias errors.

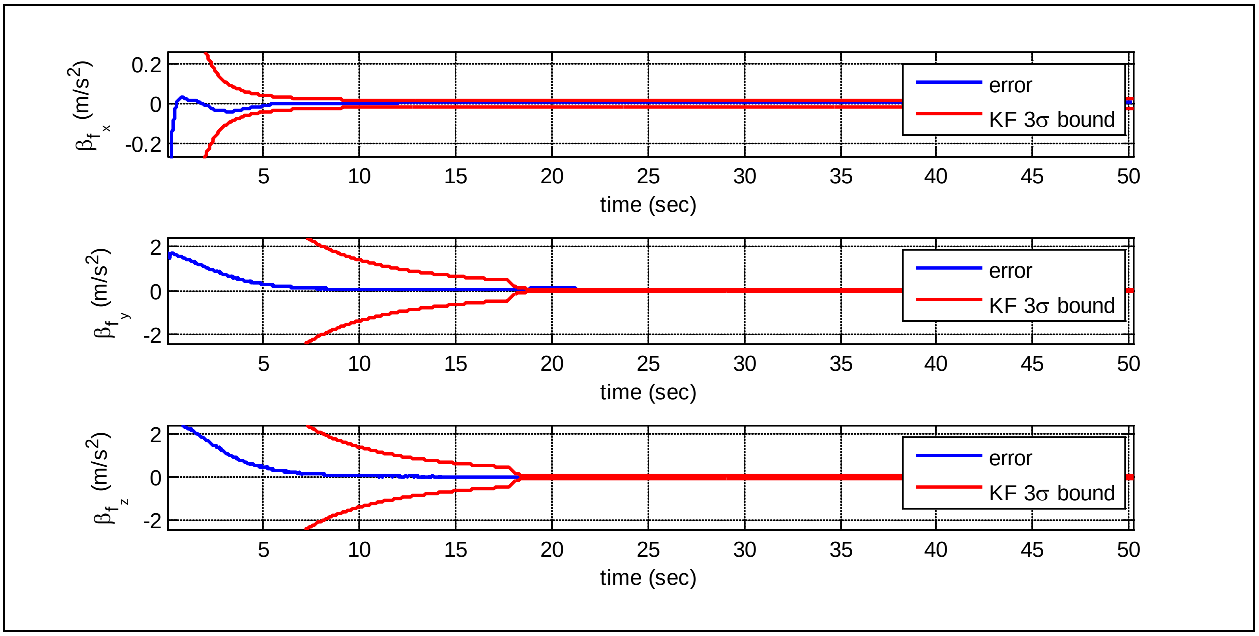

Figure 12. Indirect-fire accelerometer-bias errors.

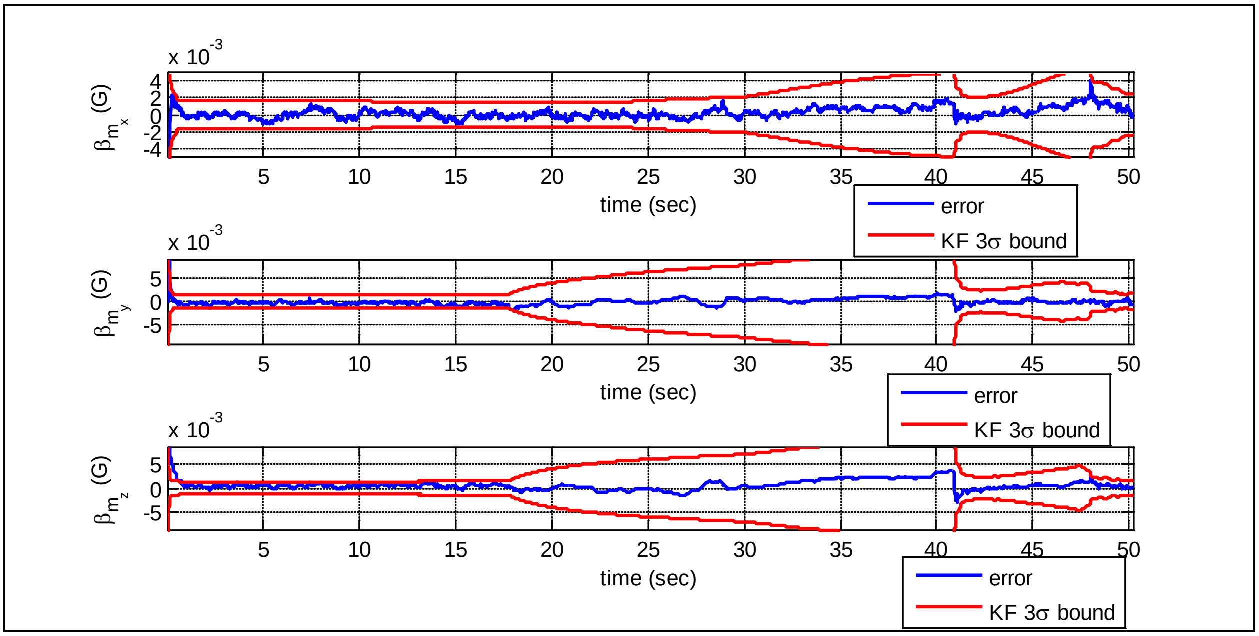

Figure 13. Indirect-fire magnetometer-bias errors.

From the covariance bounds, it is clear that some parameters are more observable while the round is spinning, while others are less so. Magnetometer biases and angle errors are more observable before the canards deploy at 16 s. However, radial accelerometer biases and radial gyroscope biases have little to no observability while the round is spinning. This is because errors in these terms are “rolled out”; that is, their effects rotate in direction at the spin frequency and never produce a noticeable change in the outputs. Angle errors and magnetometer errors grow unbounded after the canards are deployed and heuristic measurements cease. Interestingly, when the round performs a course correction maneuver at 41 s, the movement provides information about the magnetometer bias.

9. Conclusions and Future Work

The basics of multiplicative quaternion error state modeling have been presented, and some methods of using error state models in extended Kalman filters have been discussed. The main advantage of using quaternions for projectile attitude estimation is that the error state propagation equations depend only on the estimated gyroscope outputs, and not the current value of the quaternion. This provides a lot of flexibility in dealing with delayed measurements and processing time delays. The velocity and position error states are less convenient because they do depend on the current attitude estimate, but they provide a mapping between attitude errors and position and velocity errors. This in theory provides some observability into the attitude errors from the information provided from position or velocity measurements.

The measurement equations for common projectile measurements and heuristics have been presented, as well as the mappings between error states and measurement residuals. These mappings are linear, but depend on the current value of the attitude estimate. Because of this, extended Kalman filters have problems with underestimating the error state covariance particularly before the filter has converged, or when the system is not strongly observable.

The basic functionality of the error state modeling has been shown through simulation results. The direct-fire simulation required a brief heuristic yaw measurement to stabilize the attitude solution. The magnetometer bias errors, gyroscope bias errors, and attitude errors were bounded, but the filter was slightly over confident. Nevertheless, it was demonstrated that reasonable attitude estimation could be achieved with an update rate of only 30 Hz. The indirect-fire simulation demonstrated that magnetometer biases are more observable while the round is spinning, but radial accelerometer and gyroscope biases are difficult to observe until the round is despun. This doesn’t necessarily prevent someone from incorporating heuristic measurements for estimating these biases, but they are not implicitly observable through the error state modeling.

There is still a great deal of additional work to be done to develop a functional aided inertial navigation system for any real system. For example, biases may not be the dominant sensor error term, and more involved error modeling may be necessary. GPS measurements typically have their own error models that would need to be included in the filter. There is also much room for improvement in dealing with delayed measurements. One improvement for attitude estimators would be to recursively update a total state transition matrix at a higher update rate than the measurements, and use this for more accurate error state and covariance propagation. Computation time could be decreased significantly for the error state and covariance prediction steps due to matrix partitioning, making this a viable option in many cases. Decoupling of the attitude and position/velocity states may lead to better results depending on the update rates of the available measurements.

10. References

1. Fresconi, F.; Celmins, I.; Fairfax, L. *Optimal Parameters for Maneuverability of Affordable Precision Munitions*; ARL-TR-5647; U.S. Army Research Laboratory: Aberdeen Proving Ground, MD, 2011.

2. Don, M.; Gzybowski, D.; Christian, R. Roll Angle Estimation Using Thermopiles for a Mortar Flight Experiment. *62nd Aeroballistic Range Association Annual Meeting* [DVD-ROM], Nasa Glenn Research Center: Cleveland, OH, 2011.

3. Rogers, J.; Costello, M. A Low-Cost Orientation Estimator for Smart Projectiles Using Magnetometers and Thermopiles *Navigation* **Spring 2012**, *59* (1), 9–24.

4. Maley, J. Roll Orientation From Commercial Off-The-Shelf (COTS) Sensors in the Presence of Inductive Actuators. In *Institute of Navigation Joint Navigation Conference*, Colorado Springs, 2011.

5. Changey, S.; Beauvois, D.; Fleck, V.; A Mixed Extended-Unscented Filter for Attitude Estimation With Magnetometer Sensor. *American Control Conference*, 14–16 June 2006.

6. Rogers, J.; Costello, M.; Harkins, T.; Hamaoui, M. Effective Use of Magnetometer Feedback for Smart Projectile Applications. *Navigation*, **2011**, *58* (3), 203–220.

7. Fairfax, L.; Fresconi, F. *Position Estimation for Projectiles Using Low-cost Sensors and Flight Dynamics*; ARL-TR-5994; U.S. Army Research Laboratory: Aberdeen Proving Ground, MD, 2012.

8. Titterton, D. H.; Weston, J. L. *Strapdown Inertial Navigation Technology*, 2nd ed.; Progress in Astronautices and Aeronautics, Volume 207; American Institute of Aeronautics and Astronautics (AIAA): Reston, VA, 2004.

9. Toda, N. F.; Heiss, J. L.; Schlee, F. H. SPARS: The System, Algorithm, and Test Results. *Proceedings of the Symposium on Spacecraft Attitude Determination*; Report TR-0066 (5306)-12; Aerospace Corp., September–October 1969; pp 361–370, Vol.1.

10. Crassidis, J. L.; Markley, F. L.; Cheng, Y. A Survey of Nonlinear Attitude Estimation Methods. *Journal of Guidance Control and Dynamics;* American Institute of Aeronautics and Astronautics (AIAA): Reston, VA, **2007**, *30* (1), 12–28.

11. Crassidis, J. L.; Junkins, J. L. *Optimal Estimation of Dynamic Systems*; Chapman & Hall/CRC: Boca Raton, FL, 2004.

12. Roumeliotis, S. I.; Sukhatme, G. S.; Bekey, G. A. Smoother Based 3D Attitude Estimation for Mobile Robot Localization. *IEEE International Conference on Robotics and Automation* **1999**, *3*, 1979–1986.

13. Stevens, B.; Kerce, C. Guidance Navigation and Control. *Georgia Tech Professional Education: Guidance, Navigation, and Control: Theory and Applications Class Notes*, Georgia Instititue of Technology: Atlanta, GA, 2008.

14. Kuipers, J. *Quaternions and Rotation Sequences*; Princeton University Press: Princeton, NJ, 1999.

15. Bortz, J. E. A New Mathematical Formulation for Strapdown Inertial Navigation. *IEEE Transactions on Aerospace and Electronic Systems* **1971**, AES-7 (1).

16. Savage, P. G. *Strapdown Analytics*; Strapdown Associates Inc.: Maple Plain, MN, 2000.

17. Simon, D. *Optimal State Estimation*; John Wiley & Sons Inc.: Hoboken, NJ, 2006.

18. Gelb, A. *Applied Optimal Estimation*; MIT press: Cambridge, MA,1974.

19. Markley, F. L. Multiplicative vs. Additive Filtering for Spacecraft Attitude Determination. *Dynamics and Control of Systems and Structures in Space*; NASA's Goddard Space Flight Center: Greenbelt, MD, 2004.

20. Fairfax, L.; Allik, B. Vision-Based Roll and Pitch Estimation in Precision Projectiles. *AIAA Guidance Navigation and Control Conference*, Minneapolis, MN, 13–16 August 2012.

21. Murphy, C. H. *Free Flight Motion of Symmetric Missiles*; BRL-1216; U.S. Army Ballistic Research Laboratory: Aberdeen Proving Ground, MD, July 1963.

22. Uselton, B. L.; Uselton, J. C. *Test Mechanism for Measuring Pitch-Damping Derivatives of Missile Configurations at High Angles of Attack*; AEDC-TR-75-43; Arnold Engineering Development Center: Arnold AFB, TN, May 1975.

23. Jenke, L. M. *Experimental Roll-Damping, Magnus, and Static-Stability Characteristics of Two Slender Missile Configurations at High Angles of Attack (0-90 deg) and Mach Numbers 0.2 through 2.5*; AEDC-TR-76-58; Arnold Engineering Development Center: Arnold AFB, TN, July 1976.

24. Dupuis, A. *Aeroballistic Range and Wind Tunnel Tests of the Basic Finner Reference Projectile from Subsonic to High Supersonic Velocities*; TM 2002-136; Defense R&D: Valcartier, Canada, October 2002.

25. Fresconi, F. U.S. Army Research Laboratory: Aberdeen Proving Ground, MD. Personal communication, January 2013.

Appendix A. Projectile Center of Gravity Offset

Given the location of the accelerometer w.r.t an inertial frame $\mathbf{r}_a^i$, and the location of the projectile center of gravity (CG) in body fixed coordinates $\mathbf{r}_{cg}^b$, the location of the CG in inertial frame coordinates can be expressed as:

$$\mathbf{r}_{cg}^i = \mathbf{r}_a^i + \mathbf{C}_b^i \mathbf{r}_{cg}^b . \tag{A-1}$$

The output of the accelerometer will be proportional to the acceleration of the accelerometer relative to the inertial reference frame, but the projectile dynamics are usually formulated relative to the CG. The CG acceleration can be found by double differentiating equation A-1. The position of the CG in body coordinates is assumed to be constant, and from here on $\boldsymbol{\omega}_{ib}^b$ will be denoted as $\boldsymbol{\omega}$:

$$\dot{\mathbf{r}}_{cg}^i = \dot{\mathbf{r}}_a^i + \dot{\mathbf{C}}_b^i \mathbf{r}_{cg}^b + \mathbf{C}_b^i \dot{\mathbf{r}}_{cg}^b = \dot{\mathbf{r}}_a^i + \mathbf{C}_b^i \left[\boldsymbol{\omega}\times\right] \mathbf{r}_{cg}^b . \tag{A-2}$$

$$\mathbf{v}_{cg}^i = \mathbf{v}_a^i + \mathbf{C}_b^i \left[\boldsymbol{\omega}\times\right] \mathbf{r}_{cg}^b . \tag{A-3}$$

$$\dot{\mathbf{v}}_{cg}^i = \dot{\mathbf{v}}_a^i + \mathbf{C}_b^i \left[\boldsymbol{\omega}\times\right]\left[\boldsymbol{\omega}\times\right] \mathbf{r}_{cg}^b + \mathbf{C}_b^i \left[\dot{\boldsymbol{\omega}}\times\right] \mathbf{r}_{cg}^b . \tag{A-4}$$

The reaction force of the restraining “springs” against the proof mass will be proportional to the accelerometer’s acceleration with respect to the inertial reference frame minus the acceleration due to gravity. The reaction force divided by the proof mass is the specific force $\mathbf{f}_r^i$.

$$\mathbf{f}_r^i = \frac{1}{m}\left(m\dot{\mathbf{v}}_a^i - m\mathbf{g}^i\right) = \dot{\mathbf{v}}_{cg}^i - \mathbf{C}_b^i \left[\boldsymbol{\omega}\times\right]\left[\boldsymbol{\omega}\times\right] \mathbf{r}_{cg}^b - \mathbf{C}_b^i \left[\dot{\boldsymbol{\omega}}\times\right] \mathbf{r}_{cg}^b - \mathbf{g}^i . \tag{A-5}$$

The output of the accelerometer is the specific force vector resolved in the body frame:

$$\mathbf{f}^b = \mathbf{C}_i^b \mathbf{f}_r^i = \mathbf{C}_i^b \dot{\mathbf{v}}_{cg}^i - \left[\boldsymbol{\omega}\times\right]\left[\boldsymbol{\omega}\times\right] \mathbf{r}_{cg}^b - \left[\dot{\boldsymbol{\omega}}\times\right] \mathbf{r}_{cg}^b - \mathbf{C}_i^b \mathbf{g}^i . \tag{A-6}$$

Appendix B. Average Quaternion

Given an old quaternion $\mathbf{q}_1$ and a new quaternion $\mathbf{q}_2$, one way to compute the average quaternion $\bar{\mathbf{q}}$ is to compute the rotation vector $\boldsymbol{\mu}$ that transforms $\mathbf{q}_1$ into $\mathbf{q}_2$ and use the same vector with half of the magnitude to transform $\mathbf{q}_1$ to $\bar{\mathbf{q}}$. If the quaternion $\mathbf{r}$ is defined by $\mathbf{q}_2 = \mathbf{q}_1 \otimes \mathbf{r}$, then it can be solved for by $\mathbf{r} = \mathbf{q}_1^{-1} \otimes \mathbf{q}_2$. The relationship between a quaternion and a rotation vector can be used to solve for the rotation vector $\boldsymbol{\mu}$:

$$\begin{Bmatrix} r_1 \\ \mathbf{r}_{2:4} \end{Bmatrix} = \begin{Bmatrix} \cos\left(\|\boldsymbol{\mu}\|/2\right) \\ \dfrac{\boldsymbol{\mu}}{\|\boldsymbol{\mu}\|}\sin\left(\|\boldsymbol{\mu}\|/2\right) \end{Bmatrix}. \qquad \text{(B-1)}$$

$$\|\boldsymbol{\mu}\| = \left|2\cos^{-1}\left(r_1\right)\right|. \qquad \text{(B-2)}$$

$$\boldsymbol{\mu} = \mathbf{r}_{2:4}\frac{\|\boldsymbol{\mu}\|}{\sin\left(\|\boldsymbol{\mu}\|/2\right)}. \qquad \text{(B-3)}$$

The rotation vector can be thought of as the integration of and average angular velocity vector $\bar{\boldsymbol{\omega}}$ over a small time interval Δt; $\boldsymbol{\mu} = \bar{\boldsymbol{\omega}}\Delta t$. Therefore, dividing the rotation vector by 2 has the physical interpretation of integrating the angular velocity for half the time. Therefore, the new rotation quaternion $\mathbf{r}_n$ is:

$$\mathbf{r}_n = \begin{Bmatrix} \cos\left(\|\boldsymbol{\mu}\|/4\right) \\ \dfrac{\boldsymbol{\mu}}{\|\boldsymbol{\mu}\|}\sin\left(\|\boldsymbol{\mu}\|/4\right) \end{Bmatrix}. \qquad \text{(B-4)}$$

The average quaternion is then $\bar{\mathbf{q}} = \mathbf{q}_1 \otimes \mathbf{r}_n$.

NO. OF	
COPIES	ORGANIZATION
1 (PDF)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA
1 (PDF)	DIRECTOR US ARMY RESEARCH LAB IMAL HRA
1 (PDF)	DIRECTOR US ARMY RESEARCH LAB RDRL CIO LL
1 (PDF)	GOVT PRINTG OFC A MALHOTRA
1 (PDF)	RDRL WML F J MALEY

INTENTIONALLY LEFT BLANK.